5. リモート・プログラミング

5.1 インタフェースの使用方法

本器は GPIB、USB インタフェースが装備されています。
またオプション (04) で BCD インタフェースも用意されています。ただし、同時に使用することはできません。いずれかを選択して使用してください。

5.1.1 インタフェースの選択

インタフェースの選択は、正面パネルのメニューからのみ設定できます。

1. 選択したインタフェースは不揮発生メモリに保存され、電源をオフしたりインタフェースをリセットしても変わりません。
2. インタフェースには機器固有のアドレスを設定します。USB インタフェースにおいても複数の機器を接続した場合、おのおのを識別するためアドレス (USB.ID) を設定します。

インタフェースの設定項目と工場出荷状態を以下に示します。

設定項目	工場出荷状態
インタフェース選択	USB
GPIB アドレス／ USB.ID	1
GPIB リスン・オンリ機能	OFF（アドレサブル）

インタフェース選択はメニュー画面より ***I/F* → *I/F BUS*** にて使用するインタフェースを選択します。

	操作	文字表示部
1.	6146 の場合 **MENU** を押し、 , または で **..7.. *I/F*** を選択します。	MENU ..7.. I/F
	6156 の場合 **MENU** を押し、 , または で **..8.. *I/F*** を選択します。	MENU ..8.. I/F
2.	にて選択階層に移動します。	I/F GPIB BUS
3.	, または で **BUS** を選択します。 （現在設定状態）	I/F GPIB BUS
4.	にて入力・実行階層に移動します。	**BUS** **GPIB**

5. , またはで **GPIB**、**USB** または **BCD** を選択します。

	BUS
	USB

(BCD が選択できるのは、オプション 04 の BCD が搭載されている場合のみです。)

6. **EXIT** を押し、メニューを終了します。

5.1.2 互換性について

本器は、従来機種の 6144 のコマンドで動作させせることもできます。また、6144 互換モードに設定することによりさらに互換性を高めることができます。

6144 互換モードでは互換性を高めるため一部の機能制限があります。

6146/6156 の機能を最大に使用するためにも、特別に互換性を重視する場合以外は通常モードの使用を推奨します。

互換性については以下も参照してください。

5.7 発生値のデータ出力形式（トーカ・フォーマット）

5.8.3 リモート・コマンド一覧

5.8.4 6144 互換リモート・コマンド一覧

6144 互換モードの設定

	操作	文字表示部
1.	**6146** の場合 **MENU** を押し、, またはで ***..7.. I/F*** を選択します。	MENU ..7.. I/F
	6156 の場合 **MENU** を押し、, またはで ***..8.. I/F*** を選択します。	MENU ..8.. I/F
2.	にて選択階層に移動します。	I/F GPIB BUS
3.	, またはで ***6144*** を選択します。 （現在設定状態）	I/F OFF 6144
4.	にて入力・実行階層に移動します。	**6144** Ent **OFF**

6144	
Ent	ON

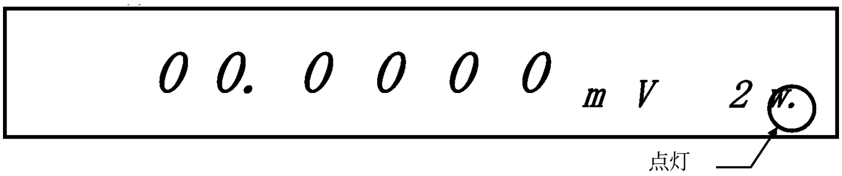
5. ◁, ▷ また ◯ で ON を選択します。
   ENTER を押し設定を実行します。設定が終了したら選択階層に移動します。
6. **EXIT** を押し、メニューを終了します。

互換モードを変更した場合、パラメータは初期化されます。初期化されるパラメータは「表 2-2 設定パラメータの初期値」を参照してください。

5.1.2.1　6144 互換モードの表示と機能の違い

6144 互換モードを設定すると、HOME 画面において、図のように表示器の右端の下部にドット（点）が点灯します。

6156 の例



6144 互換モードを ON にした場合の機能の違いを以下に示します。

- ステータス・バイト構造が 6144 互換になる。
- パラメータの初期値（C, *RST 実行時）が 6144 互換になる。
- トーカ・フォーマットが 6144 互換になる。
- 使用可能なメモリ番号が 0 ~ 159 になる。
- 一部のコマンドが使用可能または使用不可能になる。
  以下のコマンド以外は、互換モードでも使用可能です。

6144 互換モードのみで使用できるコマンド

コマンド	機能
T1	ホールド・スキャンのメモリ番号をインクリメントする
T2	シングル・スキャンの開始
T3	リピート・スキャンの開始
T?	現在実行中の（最後に実行した）スキャン・モードのクエリ
C1	スキャンの中止
C2	スキャンの一時停止

6144 互換モードで使用できないコマンド

コマンド	機能
*SRE	サービス・リクエスト・イネーブル・レジスタの設定
*SRE?	サービス・リクエスト・イネーブル・レジスタのクエリ
*ESR?	スタンダード・イベント・ステータス・レジスタ (ESR) のクエリ
*ESE	スタンダード・イベント・ステータス・イネーブル・レジスタの設定
*ESE?	スタンダード・イベント・ステータス・イネーブル・レジスタのクエリ
DSR?	デバイス・イベント・ステータス・レジスタ（DSR）のクエリ
DSE	デバイス・イベント・ステータス・イネーブル・レジスタの設定
DSE?	デバイス・イベント・ステータス・イネーブル・レジスタのクエリ
*CLS	ステータスのクリア

6144 互換モードで機能が変わるコマンド

コマンド	機能
D?	発生値のクエリ 6156 で互換モードに設定した場合、最小分解能の数値は読み出されません。
SC	メモリ・スキャンのファースト番号、ラスト番号の設定 互換モードでは使用可能なメモリ番号が 0 ~ 159 となります。
SC?	メモリ・スキャンのファースト番号、ラスト番号のクエリ 互換モードではファースト番号とラスト番号の間のセパレータがスペースとなります。
C	デバイス・クリア 互換モードではデバイス・クリアとパラメータおよびメモリの初期化を行います。
D±data[unit]	6146 で互換モードに設定した場合、30V レンジの設定で最小分解能に奇数を設定すると偶数に変換されます。 オートレンジで発生値を設定した場合、レンジを自動的に選択する範囲が 6144 と同じになります。 詳細は 6144 の取扱説明書を参照してください。
	互換モードでカウント・フィクス ON の場合、電圧／電流発生値を切換えると、発生値のカウント値が電圧値と電流値で同じ値となります。 互換モードではカウント・フィクス ON が初期値です。

6144 互換モードにおける動作の違いの詳細は、以下の項目を参照してください。

5.7 発生値のデータ出力形式（トーカ・フォーマット）

5.8.3 リモート・コマンド一覧

5.8.4 6144 互換リモート・コマンド一覧

5.2 リモート・コマンド・インデックス

このリモート・コマンド・インデックスは、5 章のリモート・コマンド索引として活用してください。

リモート・コマンド	参照ページ
*CLS	5-37
*ESE	5-37
*ESE?	5-37
*ESR?	5-37
*IDN?	5-38
*OPC	5-37
*OPC?	5-37
*OPT?	5-39
*PSC?	5-38
*PSC<n>	5-38
*RCL0	5-40
*RCL1	5-40
*RCL2	5-40
*RCL3	5-40
*RST	5-38
*SAV0	5-39
*SAV1	5-39
*SAV2	5-39
*SAV3	5-39
*SRE	5-37
*SRE?	5-37
*STB?	5-37
*TRG	5-35, 5-36
*TST?	5-38
*WAI	5-37
B	5-44
B?	5-44
BY?	5-33
BY0	5-33
BY1	5-33
C	5-38, 5-44
C0	5-44
C1	5-44
C2	5-44
C3	5-43
CAL?	5-40
CAL0	5-40
CAL1	5-40
CDN0	5-33
CDN1	5-33
CDN2	5-33
CDN3	5-33
CDN4	5-33
CDN5	5-33
CFIX?	5-32
CFIX0	5-32
CFIX1	5-32
CMP?	5-39
CMP0	5-39
CMP1	5-39
CRP?	5-39
CRP0	5-39
CRP1	5-39
CUP0	5-33
CUP1	5-33
CUP2	5-33
CUP3	5-33
CUP4	5-33
CUP5	5-33
D?	5-42
DL?	5-37
DL0	5-37
DL1	5-37
DL2	5-37
DL3	5-37
DSE	5-37
DSE?	5-37
DSR?	5-37
D±data [unit]	5-42
E	5-42
E?	5-42
ERR?	5-38
H	5-42
H?	5-42
I?	5-42
I1	5-42
I2	5-42
I3	5-42
IF	5-30
IF?	5-30
JSV?	5-31
JSV0	5-31
JSV1	5-31
K0	5-43
K1	5-43
K2	5-43

K3 5-43
K4 5-43
K5 5-43
K6 5-43
K7 5-43
LMI data1[,data2] 5-32
LMI? 5-32
LMV data1[,data2] 5-32
LMV? 5-32
MD? 5-29
MD0 5-29
MD2 5-29
MD4 5-29
N? 5-43
N?nnn 5-34
Nnnn 5-34, 5-43
NP? 5-34
NZ? 5-38
NZ0 5-38
NZ1 5-38
O? 5-44
OP? 5-39
OP0 5-39
OP1 5-39
OP2 5-39
OP3 5-39
OP4 5-39
OPH? 5-39
OPR 5-29
OPR? 5-29
P 5-34
P? 5-43
PLM? 5-33
PLM0 5-33
PLM1 5-33
RB? 5-36
RB0 5-36
RB1 5-36
RCLM? 5-35
RCLMnnn 5-35
RCLR 5-34
RINI 5-40
RLOD 5-34
ROD? 5-39
ROD0 5-39
ROD1 5-39
RS? 5-32
RS0 5-32
RS1 5-32
RSAV 5-34
S? 5-37
S0 5-37
S1 5-37
SB ±data 5-36
SB? 5-36
SBY 5-29
SBY? 5-29
SC? 5-35, 5-43
SC[mmm,]nnn 5-35, 5-43
SCM? 5-35, 5-36
SCM0 .... 5-35, 5-36
SCM1 .... 5-35, 5-36
SCM2 .... 5-35, 5-36
SI? 5-43
SIddd 5-43
SINI 5-39
SIR? 5-30
SIR1 5-30
SIR2 5-30
SIR3 5-30
SIRX? 5-30
SIRX0 5-30
SIRX1 5-30
SN ±start, ±stop, ±step 5-35
SN? 5-35
SOI ±data [unit] 5-31
SOI? 5-31
SOV ±data [unit] 5-31
SOV? 5-31
SP? 5-35, 5-36
SPdd.dd 5-35, 5-36
SRC? 5-33
SUS 5-29
SUS? 5-29
SUZ? 5-29
SUZ0 5-29
SUZ1 5-29
SVR? 5-30
SVR2 5-30
SVR3 5-30
SVR4 5-30
SVR5 5-30
SVRX? 5-30
SVRX0 5-30
SVRX1 5-30
SWSP 5-35, 5-36
T? 5-44
T1 5-44
T2 5-44
T3 5-44
TC? 5-30
TC1 5-30

TC2 5-30
TC3 5-30
TC4 5-30
TC5 5-30
TC6 5-30
TC7 5-30
TC8 5-30
TCO? 5-32
TCO0 5-32
TCO1 5-32
TF 5-30
TF? 5-30
TMC ±data 5-32
TMC? 5-32
TMP ±data 5-31
TMP? 5-31
UZ? 5-38
UZ0 5-38
UZ1 5-38
V? 5-42
V2 5-42
V3 5-42
V4 5-42
V5 5-42
V6 5-42
VF 5-30
VF? 5-30
X? 5-44
XCLR 5-41
XDT±ddddE±d 5-41
XF? 5-40
XF0 5-40
XF1 5-40
XF2 5-40
XF3 5-40
XF4 5-40
XINI 5-41
XK1 5-41
XK10 5-41
XK11 5-41
XK12 5-41
XK2 5-41
XK3 5-41
XK4 5-41
XK5 5-41
XK6 5-41
XK7 5-41
XK8 5-41
XK9 5-41
XM? 5-41
XM0 5-41
XM1 5-41
XN? 5-41
XN0 5-41
XN1 5-41
XR? 5-40
XRn 5-40
XS? 5-41
XS0 5-41
XS1 5-41
XWR 5-41

5.3　GPIB

5.3.1　概要

GPIB(General Purpose Interface Bus) を用いると、本器の発生値の設定、パラメータの設定およびデータの読み込みが外部制御できるので、自動計測システムが容易に構成できます。

本器からの GPIB 信号は、本体の発生信号系とは電気的にアイソレートされているので、外部接続機器による発生値への影響は生じません。
リモート・コマンドは USB と共通です。

- 一般仕様
  - 規格：　IEEE-488.2
  - 使用コード：　ASCII コード
  - 論理レベル：　論理 0"High" 状態　+2.4 V 以上
    論理 1"Low" 状態　+0.4 V 以下

表 5-1　インタフェース機能

コード	ファンクション
SH1	ソース・ハンドシェーク機能
AH1	アクセプタ・ハンドシェーク機能
T6	基本的トーカ機能、リスナ指定によるトーカ解除機能、シリアル・ポール機能
L3	基本的リスナ機能、トーカ指定によるリスナ解除機能、リスン・オンリ・モード機能
SR1	サービス要求機能
RL1	リモート／ローカル切り替え機能
PP0	パラレル・ポール機能なし
DC1	デバイス・クリア機能（SDC, DCL コマンドが使用できる）
DT1	デバイス・トリガ機能（GET コマンドが使用できる）
C0	コントローラ機能なし
E2	3 ステート・バス・ドライバ使用

5.3.2 GPIB 使用上の注意事項

1. 測定器との接続ケーブルや、コントローラなどと接続するバス・ケーブルは、必要以上に長くしないでください。ケーブルは 20m を超えないように注意してください。なお、弊社では標準バス・ケーブルとして以下のケーブルを用意しています。

表 5-2　標準バス・ケーブル

長さ	名称
0.5m	408JE-1P5
1m	408JE-101
2m	408JE-102
4m	408JE-104

2. バス・ケーブルのコネクタは、ピギバック形で、1 個のコネクタに male, female の両方があり、重ねて使用できます。
   バス・ケーブルを接続する場合は、3 個以上のコネクタを重ねて使用しないでください。また、コネクタ止めねじで確実に固定してください。
3. 各構成機器の電源条件、接地状態、また必要に応じて設定条件などを確認してから、各構成機器の電源を投入してください。
   バスに接続されているすべての機器の電源は、必ず ON にしてください。もし、電源を ON にしていない機器があると、システム全体の動作は保証しかねます。
4. ケーブルの着脱

   GPIB ケーブルを着脱する前に、接続の機器はすべて電源を OFF にしてください。また、各接続の筐体アースが相互に接続接地されている状態で着脱してください。
5. メッセージ転送中の ATN 割り込み

   デバイス間のメッセージ転送途中に ATN 要求が割り込んできた場合、ATN を優先して以前の状態はクリアされます。
6. プログラム・コマンドの 1 回の転送は、最大 255 文字認識します。
   プログラム・コマンドが 255 文字を超えた場合は、エラーとなります。
7. プログラム・コマンド送出後、5ms 以上は REN ラインを LOW に保持してください。
8. リスン・オンリ・モードで使用する場合はコントローラは接続しないでください。
9. *TRG および E コマンドは実行が完了する前に次のコマンドを受け付けることが可能です。
   コマンド実行完了と同期を取るには＊ OPC、*OPC?、*WAI コマンドを使用してください。
   本コマンドは 1 行のプログラム行の最後に記述してください。
   例 :“ *TRG, *OPC ”、“ *TRG, *OPC? ”、“ *TRG, *WAI ”

5.3.3 GPIB の設定

下記の設定メニューは、インタフェース選択が GPIB の場合に設定可能となります。

- アドレス設定

	操作	文字表示部
1.	6146 の場合 **MENU** を押し、◁ , ▷ または ○ で ***..7.. I/F*** を選択します。	MENU ..7.. I/F
	6156 の場合 **MENU** を押し、◁ , ▷ または ○ で ***..8.. I/F*** を選択します。	MENU ..8.. I/F
2.	▽ にて選択階層に移動します。 （現在設定の I/F） （現在の I/F 設定が GPIB の場合は 6 項から操作してください。）	I/F USB BUS
3.	▽ にて入力・実行階層に移動します。	BUS **USB**
4.	◁, ▷ または ○ で **GPIB** を選択します。	BUS **GPIB**
5.	**ENTER** にて確定します（選択階層に移動します）。 （現在設定の I/F）	I/F GPIB BUS
6.	◁, ▷ または ○ で ***GP.Adr***（GPIB アドレス）を選択します。 （現在設定の GPIB アドレス）	I/F 01 GP.Adr
7.	▽ にて入力・実行階層に移動します。	GP.Adr **01**
8.	◁, ▷ で変更桁を選択し、▽, △ または ○ で ***GP.Adr*** を設定します。	GP.Adr **02**
9.	**ENTER** にて確定します（選択階層に移動します）。 （現在設定の GPIB アドレス）	I/F 02 GP.Adr
10.	**EXIT** を押し、メニューを終了します。	

- リスン・オンリ設定

	操作	文字表示部
1.	6146 の場合 **MENU** を押し、, または で ***..7.. I/F*** を選択します。	MENU ..7.. I/F
	6156 の場合 **MENU** を押し、, または で ***..8.. I/F*** を選択します。	MENU ..8.. I/F
2.	にて選択階層に移動します。 （現在設定の I/F） （現在の I/F 設定が GPIB の場合は 6 項から操作してください。）	I/F USB BUS
3.	にて入力・実行階層に移動します。	BUS **USB**
4.	, **,** または で **GPIB** を選択します。	BUS **GPIB**
5.	**ENTER** にて確定します（選択階層に移動します）。 （現在設定の I/F）	I/F GPIB BUS
6.	, または で ***L.Only***（リスン・オンリ）を選択します。 （現在設定状態）	I/F OFF L.Only
7.	にて入力・実行階層に移動します。	L.Only **OFF**
8.	, **,** または で ON/OFF を選択します。	L.Only **ON**
9.	**ENTER** にて確定します（選択階層に移動します）。 （現在設定状態）	I/F ON L.Only
10.	**EXIT** を押し、メニューを終了します。	

5.4　USB

5.4.1　概要

本器は USB2.0 規格に準拠した USB(Universal Serial Bus) を標準装備しています。
USB を用いると、バス上の複数台の本器に対する機能の設定および測定データの読み込みが、パーソナル・コンピュータより可能となり自動計測システムが容易に構成できます。

注意　すべてのパーソナル・コンピュータ、ハブ等での動作を保証するものではありません。

5.4.2　USB 仕様

• 規格：	USB2.0 Full-Speed 準拠
• 使用コネクタ：	USB B タイプ（メス）
• 識別 ID：	USB.ID として 1 ~ 127 まで設定可能
• リモート／ローカル：	機能あり
• 入力コマンド：	ASCII 文字列コマンドによる機能設定、クエリ
• 出力フォーマット：	ASCII 文字列による測定データ、クエリ応答出力
• ドライバ：	ADC 計測器 USB ドライバ 2.3.0 以降を使用

5.4.3　USB のセットアップ

5.4.3.1　パーソナル・コンピュータとの接続

本器背面部の USB コネクタ（B タイプ）とパーソナル・コンピュータの USB コネクタを接続ケーブルで接続してください。
接続の際はコネクタを確実に最後まで挿入してください。

1 台のパーソナル・コンピュータに複数台の本器を接続する場合は、USB ハブを使用してください。

5.4.3.2　USB I/F および USB.ID の設定

USB.ID 設定メニューは、インタフェース選択が USB の場合に設定可能となります。

	操作	文字表示部
1.	6146 の場合 **MENU** を押し、, またはで ***..7.. I/F*** を選択します。	MENU ..7..　I/F
	6156 の場合 **MENU** を押し、, またはで ***..8.. I/F*** を選択します。	MENU ..8..　I/F
2.	にて選択階層に移動します。 （現在設定の I/F） （現在の I/F 設定が USB の場合は 6 項から操作してください。）	I/F GPIB　BUS
3.	にて入力・実行階層に移動します。	BUS **GPIB**
4.	, または で USB を選択します。	BUS **USB**
5.	**ENTER** にて確定します（選択階層に移動します）。 （現在設定の I/F）	I/F USB　BUS
6.	, またはで ***USB id*** を選択します。 （現在設定の USB id）	I/F 001　USBid
7.	にて入力・実行階層に移動します。	USBid 001
8.	, で変更桁を選択し、, またはで ***USB id*** の値を設定します。	USBid **001**
9.	**ENTER** にて確定します（選択階層に移動します）。 （現在設定の USBid）	I/F 002　USBid
10.	**EXIT** を押し、メニューを終了します。	

5.4.3.3　USB 使用上の注意事項

クエリ・コマンドを実行する場合、直前に実行したコマンドとの間に約 20ms の待ち時間を入れてください。

5.5　BCD（オプション 04）

5.5.1　概要

本器はオプションにて BCD パラレル・インタフェースを装備できます。
本インタフェースによって､汎用コンピュータのパラレル I/O やシーケンサなどに接続できます。

注意 **BCD リモートでは熱電対起電力出力の制御はできません。**
BCD リモートを使用する場合は外部単線入力信号「TRIGGER IN」および「INTERLOCK」､「OPERETE IN/OUT」には何もつながないでください。誤動作の原因となります。
ケーブルを脱着する前に､接続の機器は全て電源を OFF にしてください。また､各接続の筐体アースが相互に接地されている状態で脱着してください。

5.5.2　BCD インタフェースの設定

以下の設定メニューは、インタフェース選択が USB の場合に設定可能となります。

	操作	文字表示部
1.	6146 の場合 **MENU** を押し、 , または で ***..7.. I/F*** を選択します。	MENU ..7..　I/F
	6156 の場合 **MENU** を押し、 , または で ***..8.. I/F*** を選択します。	MENU ..8..　I/F
2.	にて選択階層に移動します。 （現在設定の I/F）	I/F GPIB　BUS
3.	にて入力・実行階層に移動します。	BUS **GPIB**
4.	, または で BCD を選択します。	BUS **BCD**
5.	**ENTER** にて確定します。	

BCD インタフェースが選択されると、表示部の左に "RMT" と表示されます。BCD インタフェースの解除は LOCAL キーを押します。解除されると "RMT" の表示が消えます。

注意 **BCD インタフェースが解除されると、インタフェースは BCD インタフェース選択直前の選択に戻ります。再度 BCD インタフェースを選択したい場合は、上記の操作を行ってください。**

5.5.3 インタフェース信号

ピン配列とコード表を以下に示します。

表 5-3　BCD PARALLEL コネクタのピン配列およびコード表

ピン番号	信号名		ピン番号	信号名		
1	0V (GND)		19	30V		高
2	1		20	3V		*2
3	2	10^1 桁	21	300mV	レンジ・コード	
4	4	（10^0 桁）*1	22	30mV		優先順位
5	8		23	200mA		
6	1		24	30mA		
7	2	10^2 桁	25	3mA		
8	4	（10^1 桁）*1	26	30V		*3 低
9	8		27	極性		
10	1		28	2	10^5 桁（10^4 桁）*1	
11	2	10^3 桁	29	出力 ON		
12	4	（10^2 桁）*1	30	1		
13	8		31	2	10^0 桁（NC）*1	
14	1		32	4		
15	2	10^4 桁	33	8		
16	4	（10^3 桁）*1	34	NC		*4
17	8		35	NC		
18	1	10^5 桁（10^4 桁）*1	36	LOAD		

使用コネクタ : 57LE-40360（アンフェノール社製）または相当品

*1:　6146 の場合の出力レベル設定です。

*2:　レンジ・コードが同時に論理 "1" となった場合、優先度があります。

*3:　30V レンジは OR で判定します。（ピン番号 19 に割り付けられている 30V レンジは、従来機の 6144 では 10V レンジです。）

*4:　NC (Non-Internal Connection): 中継端子として使用しないでください。

5.5.4　信号の電気的仕様

電気的仕様はすべて TTL 負論理（Lo レベルが論理 "1"、Hi レベルが論理 "0"）となっています。

論理 "1": 0V ~ +0.4V（または GND ショート）
論理 "0": +2.4V ~ +5.25V（またはオープン）

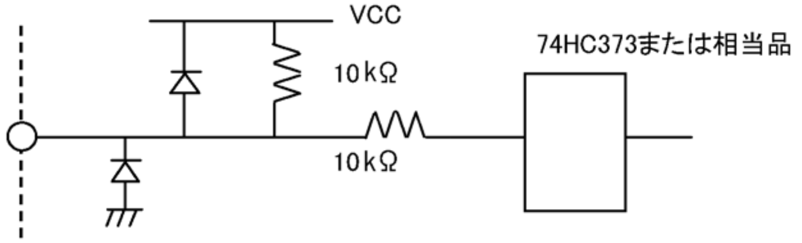


図 5-1　BCD インタフェース入力等価回路

5.5.5　設定の仕様

出力レベル設定コード :	5 桁パラレル BCD コード　全 18 ビット (6146) 6 桁パラレル BCD コード　全 22 ビット (6156)
出力レベル設定範囲 :	200mA レンジを除き ; 00000 ~ 32000(6146) / 000000 ~ 320000(6156) 200mA レンジ ; 00000 ~ 22000(6146) / 000000 ~ 220000(6156)

注意　出力設定範囲を超えた BCD コードが入力された場合、そのレンジにおける最大値（32000 / 320000 または 22000 / 220000）が設定されます。

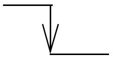
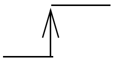
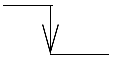
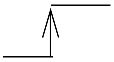
レンジ :	30V, 3V, 300mV, 30mV, 200mA, 30mA, 3mA, (30V)　全 8 ビット
極性 :	プラスまたはマイナス（論理 0 でプラス、論理 1 でマイナス） 1 ビット
出力 ON:	オペレートまたはスタンバイ （論理 0 でスタンバイ、論理 1 でオペレート）　1 ビット
LOAD:	制御信号　1 ビット

5.5.6 BCD リモート状態

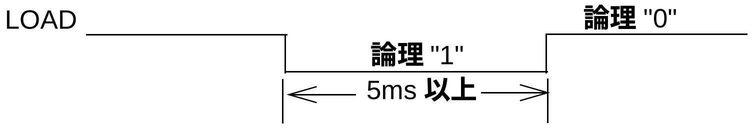
本器の BCD パラレル・インタフェースは、コントロール中はリモート状態となります。リモート状態の解除は **LOCAL** キーを押してください。**LOCAL** キーを押すと BCD リモート状態は解除され、BCD インタフェースの設定も解除となり、USB または GPIB に設定されます。

BCD リモート状態では、表示部の左サイド・ランプ部の RMT ランプが点灯します。

BCD リモート状態における LOAD 信号機能を以下に示します。

LOAD 信号	BCD リモート状態		
		出力 ON 信号	出力 ON 信号の変化に伴う動作内容
論理 "0" +2.4V ~ +5.25V またはオープン	**LOCAL** キーのみ有効 出力レベルの設定不可		レンジ、極性信号による設定が行われ、オペレートとなる
			レンジ、極性信号による設定を内部へ取り込み、スタンバイとなる
論理 "1" 0V ~ +0.4V または GND ショート	**LOCAL** キーのみ有効 レンジ、極性、出力レベル設定がリモートで可能		オペレートとなる
			スタンバイとなる

注意　LOAD **信号の論理** "1" **の区間は** 5ms **以上にしてください。**

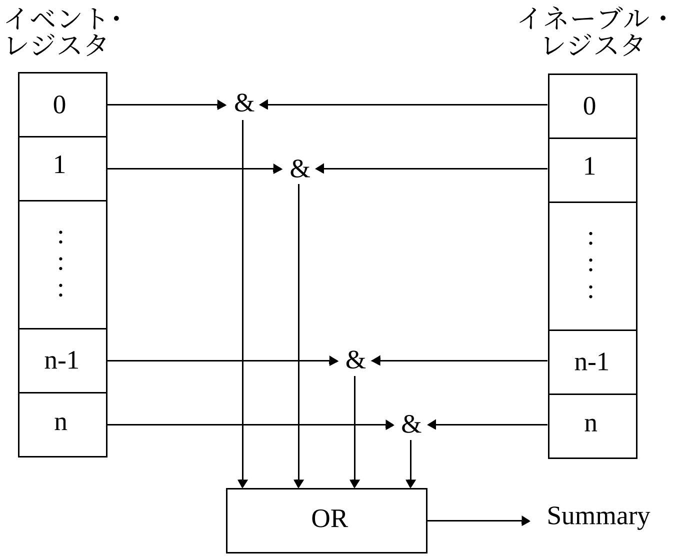


5.6 ステータス・レジスタ構造

本器では IEEE 規格 488.2-1987 に適合した階層化されたステータス・レジスタ構造をもち、機器のさまざまな状態をコントローラへ送信できます。ここではこのステータス構造の動作モデルと、イベントの割当を説明します。

5.6.1 ステータス・レジスタ

1. ステータス・レジスタ概要
   本器は、IEEE 規格 488.2-1987 で定義されたステータス・レジスタのモデルを採用し、イベント・レジスタ、イネーブル・レジスタから構成されています。



- イベント・レジスタ
  イベント・レジスタは、各イベントに応じたステータスをラッチして保持します（変化を保持する場合もある）。
  このレジスタがセットされると、クエリで読み出されるか、*CLS でクリアされるまでセットされたままです。
  イベント・レジスタにデータを書き込むことはできません。
- イネーブル・レジスタ
  イネーブル・レジスタは、イベント・レジスタのどのビットを有効なステータスとしてサマリを生成するのか指定します。イネーブル・レジスタはイベント・レジスタと AND をとられ、その結果の OR がサマリとして生成されます。サマリはステータス・バイト・レジスタに書き込まれます。
  イネーブル・レジスタはデータを書き込めます。

本器のステータス・レジスタは、以下の 3 種類があります。

- ステータス・バイト・レジスタ (STB)
- スタンダード・イベント・ステータス・レジスタ (SESR)
- デバイス・イベント・ステータス・レジスタ (DESR)

本器のステータス・レジスタの構造を図 5-2 に示します。

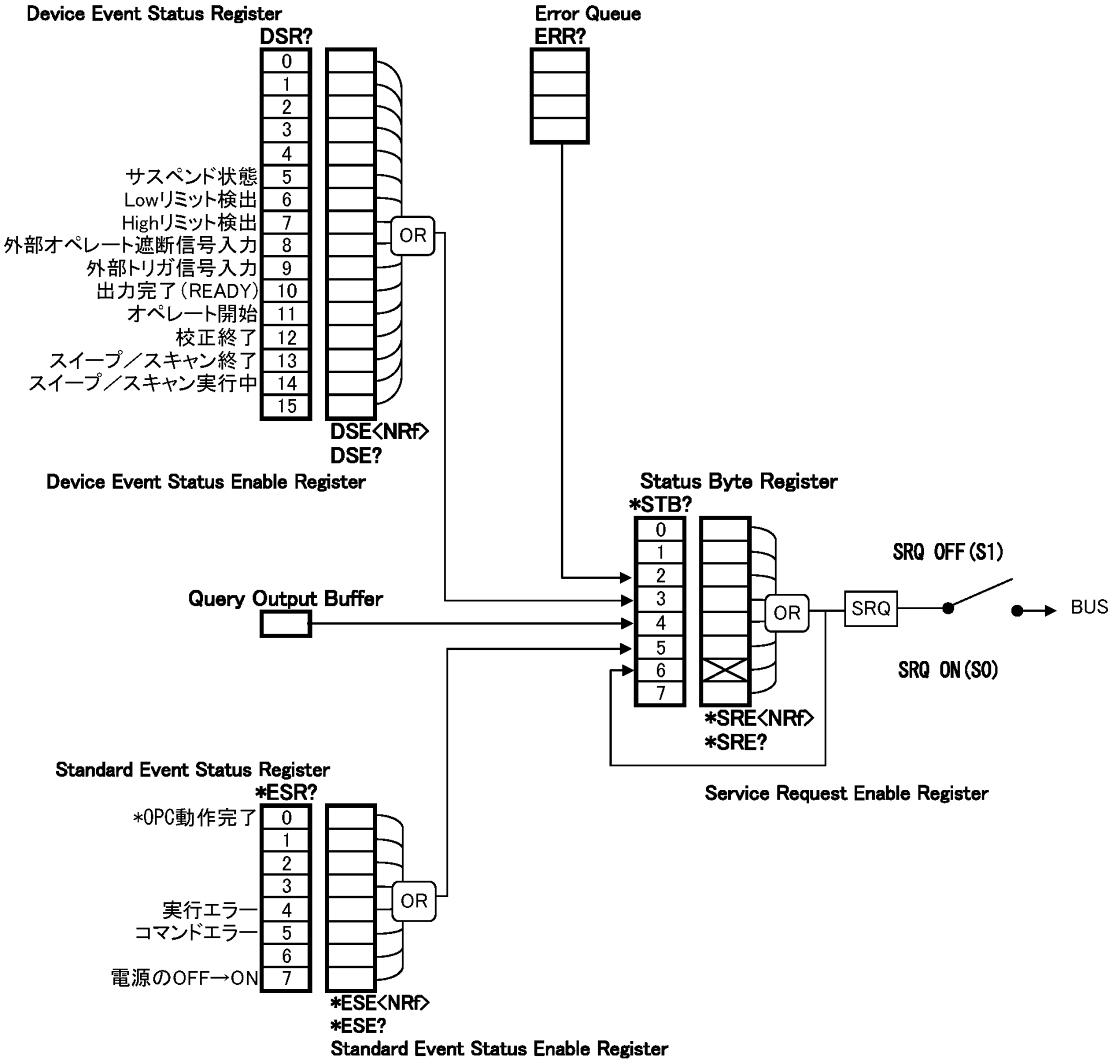


図 5-2　ステータス・レジスタの構造

2. イベント・イネーブル・レジスタ

各イベント・レジスタには、どのビットを有効にするかを決めるイネーブル・レジスタがあります。イネーブル・レジスタは、対応するビットを 10 進値で設定します。

- サービス・リクエスト・イネーブル・レジスタのセット : *SRE
- スタンダード・イベント・ステータス・イネーブル・レジスタのセット : *ESE
- デバイス・イベント・イネーブル・レジスタのセット : DSE

(例)　デバイス・イベント・レジスタの EOM ビットのみを有効にします。
デバイス・イベント・レジスタの EOM ビットが 1 にセットされると、ステータス・バイト・レジスタの DSB ビットが 1 にセットされます。

（例）　ステータス・バイト・レジスタの DSB（Device Event Status Register のサマリ）ビットと ESB（Standard Event Status Register のサマリ）ビットを有効にします。
DSB ビットまたは ESB ビットが 1 にセットされると、ステータス・バイト・レジスタの MSS ビットが 1 にセットされます。

3. ステータス・バイト・レジスタ

ステータス・バイト・レジスタは、ステータス・レジスタからの情報を要約しています。また、このステータス・バイト・レジスタのサマリがサービス・リクエストとしてコントローラに送信されます。そのため、ステータス・バイト・レジスタは、ステータス・レジスタ構造とは若干違った動作を行います。ここではステータス・バイト・レジスタに関して説明をします。
ステータス・バイト・レジスタの構造を、図 5-3 に示します。

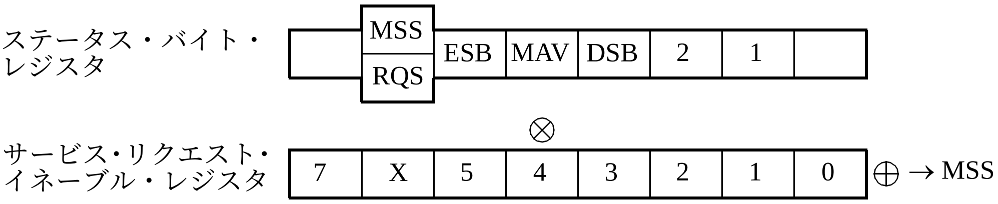


図 5-3　ステータス・バイト・レジスタの構造

このステータス・バイト・レジスタは、以下の 3 点を除くとステータス・レジスタに従います。

- ステータス・バイト・レジスタのサマリが、ステータス・バイト・レジスタの bit6 に書き込まれます。
- イネーブル・レジスタの bit6 は、常に有効で変更できません。
- ステータス・バイト・レジスタの bit6 (MSS) が、サービス・リクエスト要求の RQS を書き込みます。

このレジスタが、コントローラからのシリアル・ポールに対して応答します。シリアル・ポールに対して応答するときには、ステータス・バイト・レジスタの bit0 ~ 5、bit7 および RQS が読み出され、その後に RQS は 0 にリセットされます。その他のビットはそれぞれの要因が 0 になるまでクリアされません。
ステータス・バイト・レジスタ、RQS、MSS は、"*CLS" を実行するとクリアできます。それにともなって、SRQ ラインも偽になります。

ステータス・バイト・レジスタの各ビットの意味を表 5-4 に示します。

表 5-4　ステータス・バイト・レジスタ (STB) (1/2)

bit	名称	内容
0	未使用	常に 0
1	未使用	常に 0
2	EAV Error Available	ON : Error Queue にエラー情報が格納されたときに 1 に設定される OFF : Error Queue が読み出されて空になったときに 0 に設定される
3	DSB Device Event Summary Bit	ON : DESR のいずれかの事象が発生して 1 になったとき、DESER の対応ビットが 1 であれば、このビットが 1 に設定される OFF : DESR が読み出し (DSR?) によりクリアされたときに 0 に設定される

表 5-4　ステータス・バイト・レジスタ (STB) (2/2)

bit	名称	内容
4	MAV Message Available	ON : 出力バッファにクエリ応答データが入力されたときに 1 が設定される OFF : 出力バッファが読み取られ空になったときに 0 が設定される
5	ESB Standard Event Summary Bit	ON : SESR のいずれかの事象が発生して 1 になったとき、SESER の対応ビットが 1 であれば、このビットが 1 に設定される OFF: SESR が読み出し (*ESR?) によりクリアされたとき 0 に設定される
6	MSS Master Summary	ON : STB のいずれかの事象が発生したとき、SRER の対応ビットが 1 であれば、このビットが 1 に設定される
	RQS Request Service	ON : MSS が 1 になり、SRQ が発生すると RQS が 1 になる OFF : シリアル・ポールで STB が読み出されたとき
7	未使用	常に 0

ステータス・バイト・レジスタがクリアされる共通条件

- 電源投入ですべてクリア
- *CLS ですべてクリア、ただし出力バッファにデータがある場合は MAV はクリアしない
- *STB? で読み出してもクリアされない

サービス・リクエスト・イネーブル・レジスタがクリアされる条件

- 電源投入時（PSC フラグが 1 のとき）
- *SRE0 コマンドを実行したとき

4. スタンダード・イベント・ステータス・レジスタ
   スタンダード・イベント・ステータス・レジスタの割り当てを、表 5-5 に示します。

表 5-5　スタンダード・イベント・ステータス・レジスタ (ESR)

bit	名称	内容
0	OPC Operation Complete	ON : *OPC コマンド受信後、実行中の全動作が終了すると 1 が設定される
1	未使用	常に 0
2	未使用	常に 0
3	未使用	常に 0
4	EXE Execution Error	ON : 受信したコマンドが現在実行不可能なときに 1 が設定される コマンドのパラメータに誤りがあったときに 1 が設定される
5	CME Command Error	ON : 受信したコマンドのつづりが誤っていたときに 1 が設定される
6	未使用	常に 0
7	PON Power On	ON : 電源 OFF から ON 時に 1 が設定される

スタンダード・イベント・ステータス・レジスタがクリアされる共通条件

- 電源投入ですべてクリア
- *CLS ですべてクリア
- *ESR? で読み出すことによりすべてクリアされる

スタンダード・イベント・ステータス・イネーブル・レジスタがクリアされる条件

- 電源投入時
- *ESE0 コマンドを実行したとき

5. デバイス・イベント・ステータス・レジスタ
   デバイス・イベント・ステータス・レジスタの割り当てを表 5-6 に示します。

表 5-6　デバイス・イベント・ステータス・レジスタ (DSR) (1/2)

bit	名称	内容
0	未使用	常に 0
1	未使用	常に 0
2	未使用	常に 0
3	未使用	常に 0
4	未使用	常に 0
5	SUS Suspend	ON : サスペンド状態になったときに 1 が設定される OFF :スタンバイ／オペレート状態になったときに 0 が設定される
6	LML Limiter Low	ON : Low リミッタ検出時に 1 が設定される
7	LMH Limiter High	ON : High リミッタ検出時に 1 が設定される
8	EOP Ext.Operate Off In	ON : 外部オペレート遮断信号入力を検出したときに 1 が設定される
9	ETG Ext.Trigger In	ON : 外部トリガ信号入力を検出したときに 1 が設定される
10	RDY READY Out	ON : スタンバイ／サスペンド状態からオペレート状態に変わったとき、またはオペレート状態で発生値を変更したときから 10ms（または 50ms）後に 1 が設定される OFF : 発生値変更処理中、またはスタンバイ／サスペンド状態にしたときに 0 に設定される
11	OPR Operate	ON : オペレート状態になったときに 1 が設定される OFF :スタンバイ状態になったときに 0 が設定される
12	CAE Calibration End	ON : 校正終了のときに 1 が設定される OFF :校正開始 のときに 0 が設定される
13	SWE Sweep End	ON : スキャン／スイープ終了のときに 1 が設定される OFF :スキャン／スイープ開始 のときに 0 が設定される

表 5-6　デバイス・イベント・ステータス・レジスタ (DSR) (2/2)

bit	名称	内容
14	SWR Sweep Run	ON : スキャン／スイープのスタート時に 1 が設定される OFF : スキャン／スイープ終了および一時停止したときに 0 が設定される
15	未使用	常に 0

デバイス・イベント・ステータス・レジスタがクリアされる共通条件

- 電源投入ですべてクリア
- *CLS ですべてクリア
- DSR? で読み出すことによりすべてクリアされる

デバイス・イベント・ステータス・イネーブル・レジスタがクリアされる共通条件

- 電源投入時
- DSE0 コマンドを実行したとき

5.6.2　6144 互換モードのステータス・レジスタ

6144　互換モードに設定したときの　ステータス・レジスタ構造および、各ビットの説明を図 5-4、表 5-7 に示します。

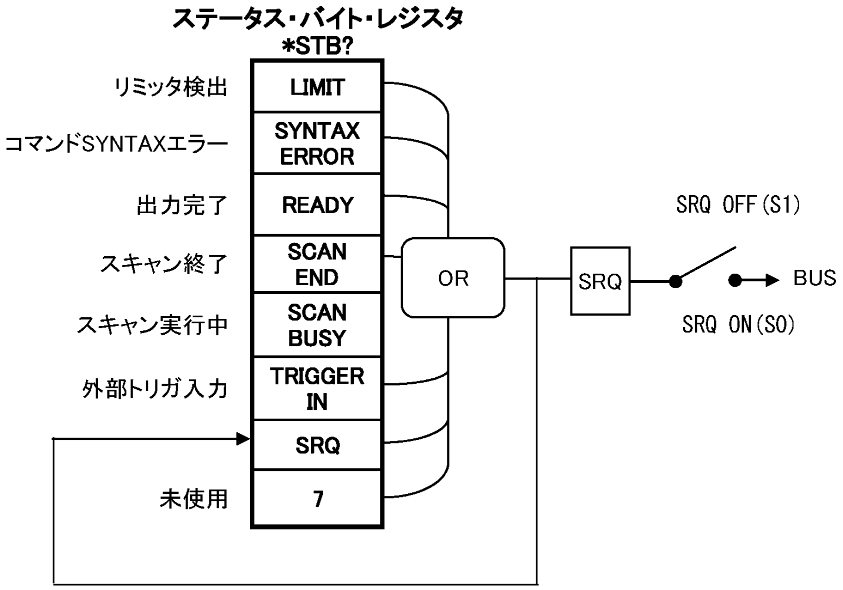


図 5-4　6144 互換モードのステータス・レジスタ構造

表 5-7　ステータス・バイト・レジスタ (STB) 6144 互換モード

bit	名称	内容
0	LIMIT	ON : Hi/Lo リミッタを検出したときに 1 が設定される OFF : リミッタが解除されると 0 に設定される シリアル・ポールで読み出したときに 0 に設定される。ただしポーリングが実行される前に解除された場合、SRQ ビットはポーリングされるまで保持される
1	SYNTAX ERROR	ON : 未認知コード、文法の誤り、設定範囲外のいずれかを処理中に検出したとき 1 が設定される エラーの検出されたコマンド以降のコマンドは無視される OFF : 正常なコマンドを受け付けたとき 0 に設定される
2	READY	ON : スタンバイ状態からオペレート状態に変わったとき、またはオペレート状態で発生値を変更したときから 50ms（または 10ms）後に 1 が設定される OFF : 発生値変更処理中や、またはスタンバイ状態にしたときに 0 に設定される シリアル・ポールで読み出したときに 0 に設定される
3	SCAN END	ON : シングル・スキャン／スイープ動作がラスト No. まで達したときに 1 が設定される OFF : 次のスキャン／スイープ開始時に 0 に設定される シリアル・ポールで読み出したときに 0 に設定される
4	SCAN BUSY	ON : スキャン／スイープが実行されている間は 1 が設定される OFF : スキャン／スイープが終了／停止されると 0 が設定される
5	TRIGGER IN	ON : コンティニュアス動作中に TRIGGER 端子からの入力信号を検出すると 1 が設定される OFF: コンティニュアス動作コマンドを受信すると 0 に設定される シリアル・ポールで読み出したときに 0 に設定される
6	SRQ	ON : bit4 を除く STB のいずれかの事象が発生したとき、このビットが 1 に設定される OFF : シリアル・ポールで STB が読み出されたとき 0 に設定される
7	未使用	常に 0

ステータス・バイト・レジスタがクリア (=0) される共通条件

- 電源投入ですべてクリア
- *STB? で読み出してもクリアされない
- シリアル・ポールで読み出したときにクリアする

5.7 発生値のデータ出力形式（トーカ・フォーマット）

発生値を読み出したときの出力形式を示します。

1. クエリ・コマンドによる読み出し

クエリ・コマンド		応答	
	レンジ	6146 の場合	6156 の場合
SOV?	30mV	SOV±dd.dddE-3	SOV±dd.ddddE-3
	300mV	SOV±ddd.ddE-3	SOV±ddd.dddE-3
	3V	SOV±d.ddddE+0	SOV±d.dddddE+0
	30V	SOV±dd.dddE+0	SOV±dd.ddddE+0
SOI?	3mA	SOI±d.ddddE-3	SOI±d.dddddE-3
	30mA	SOI±dd.dddE-3	SOI±dd.ddddE-3
	200mA	SOI±ddd.ddE-3	SOI±ddd.dddE-3
TMP?	*1	-	TMP±dddd.dE+0
D?	30mV	DV±d.ddddE-2	DV±d.dddddE-2
	300mV	DV±d.ddddE-1	DV±d.dddddE-1
	3V	DV±d.ddddE+0	DV±d.dddddE+0
	30V	DV±d.ddddE+1	DV±d.dddddE+1
	3mA	DI±d.ddddE-3	DI±d.dddddE-3
	30mA	DI±d.ddddE-2	DI±d.dddddE-2
	200mA	DI±d.ddddE-1	DI±d.dddddE-1
6144 互換モード時			
D?	30mV	DV±d.ddddE-2	
	300mV	DV±d.ddddE-1	
	3V	DV±d.ddddE+0	
	30V	DV±d.ddddE+1	
	3mA	DI±d.ddddE-3	
	30mA	DI±d.ddddE-2	
*2	200mA	DI±d.ddddE-1	

*1 : 熱起電力出力は 6156 のみの機能です。
*2 : 6156 の最小桁は出力されません。

2. クエリ・コマンドによらない読み出し

   クエリ・コマンドを送らずに本器をトーカ指定し、データを受信することにより常に現在の発生値を読み出すことができます。

- 通常モードのトーカ・フォーマット

ファンクション	レンジ	6146 の場合	6156 の場合
電圧	30mV	SOV±dd.dddE-3	SOV±dd.ddddE-3
	300mV	SOV±ddd.ddE-3	SOV±ddd.dddE-3
	3V	SOV±d.ddddE+0	SOV±d.dddddE+0
	30V	SOV±dd.dddE+0	SOV±dd.ddddE+0
電流	3mA	SOI±d.ddddE-3	SOI±d.dddddE-3
	30mA	SOI±dd.dddE-3	SOI±dd.ddddE-3
	200mA	SOI±ddd.ddE-3	SOI±ddd.dddE-3
熱起電力 *1		-	TMP±dddd.dE+0

- 6144 互換モードのトーカ・フォーマット

ファンクション	レンジ	6146 の場合	6156 の場合 *2
電圧	30mV	DV±d.ddddE-2	
	300mV	DV±d.ddddE-1	
	3V	DV±d.ddddE+0	
	30V	DV±d.ddddE+1	
電流	3mA	DI±d.ddddE-3	
	30mA	DI±d.ddddE-2	
	200mA	DI±d.ddddE-1	
熱起電力 *1		-	TMP±dddd.dE+0

*1 : 熱起電力出力は 6156 のみの機能です。
*2 : 6156 の最小桁は出力されません。

3. ブロック・デリミタ
   1 つのデータの終わりを示すためにブロック･デリミタを出力します。コマンドによりブロック・デリミタを指定することができます。

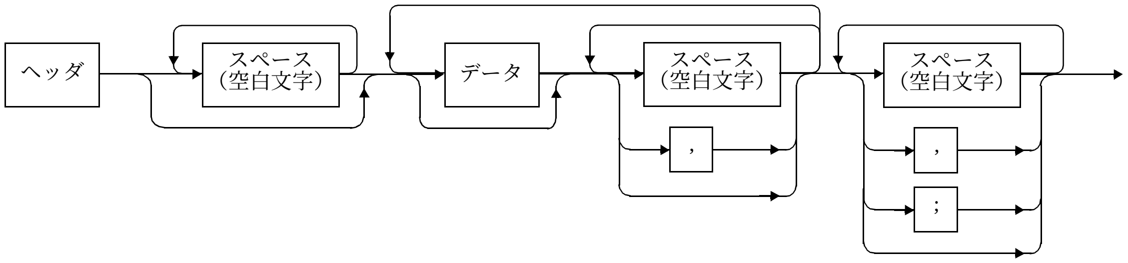
ブロック・デリミタ	設定コマンド	初期値
CR LF+EOI	DL0	●
LF	DL1	
EOI	DL2	
LF+EOI	DL3	

（EOI は GPIB の機能です。USB では出力されません。）

5.8　リモート・コマンド

5.8.1　　コマンド文法

コマンド文法は、以下のフォーマットで定義されています。



1. ヘッダ

   ヘッダには、共通コマンド・ヘッダと単純ヘッダがあります。共通コマンド・ヘッダは、ニーモニックの先頭にアスタリスク (*) を付けたものです。
   単純ヘッダは、階層構造を持たない、機能的に独立した命令です。
   ヘッダの英文字の直後に ? を付けるとクエリ・コマンドになります。

2. スペース（空白文字）

   1 文字分以上のスペースが可能です（スペースを省略しても構いません）。

3. データ

   コマンドが複数のデータを必要とするときは、データをカンマ (,) で区切って複数並べます。カンマ (,) の前後にスペース（空白文字）を入れても構いません。データ・タイプの詳細については、「5.8.2 データ・フォーマット」を参照してください。

4. 複数のコマンドの記述

   本器は、複数のコマンドを連続またはセミコロン (;)、カンマ (,)、スペース (␣) で区切って 1 行で記述することが可能です。

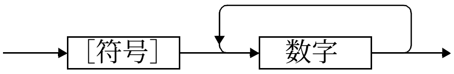
5.8.2 データ・フォーマット

本器は、ここで示すデータ・タイプをデータの入出力で使用します。

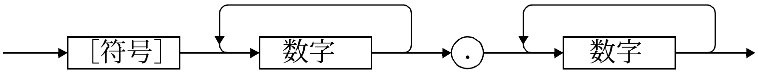
1. 数値データ

   数値データには以下の 3 つのフォーマットがあり、本器に対する数値の入力では、どれを用いても構いません。また、コマンドによっては入力時に単位を付けられます。

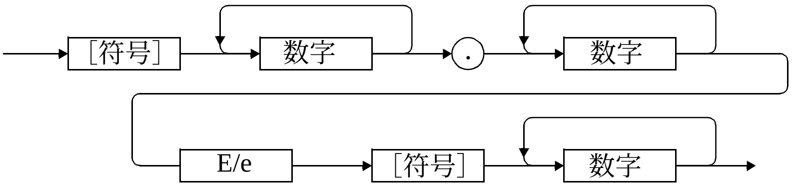
   - 整数型 : NR1 フォーマット



   - 固定小数点型 : NR2 フォーマット



   - 浮動小数点型 : NR3 フォーマット



2. 単位

   SOV, SOI, D コマンドで使用可能な単位の一覧を以下に示します。

単位	指数	意味
V	10^{0}	電圧
MV	10^{-3}	電圧
UV	10^{-6}	電圧
A	10^{0}	電流
MA	10^{-3}	電流
UA	10^{-6}	電流

5.8.3 リモート・コマンド一覧

1. 初期値の欄は、電源 ON 時、工場出荷時に初期化される状態を示します。
    - 電源 ON 時の項目は、電源投入時の状態を示します。
    - *RST および RINI コマンドでは、工場出荷時の値に初期化されます。
      ただし、*4 は RINI コマンドで、*5 は RINI および *RST コマンドで初期化されません。
2. コマンド表の記述上の注意事項
    - コマンド表の [ ] で囲んだパラメータは、省略可能なことを示しています。
    - コマンド表の < > で囲んだパラメータは、1 つのデータの区切りを示しています。
    - 動作可否の欄 は、以下のことを示しています。

      メモリ動作中：　メモリ・リコール・モードでシングル／リピート・スキャン実行中（一時停止中を含む）
      ○：動作可
      ×：動作不可

      スイープ動作中：　スイープ・モードでオペレート中
      ○：動作可
      ×：動作不可
      △：オペレート中は動作可、スイープ実行中（一時停止中を含む）は不可

分類	項目	コマンド	内容	初期値		動作可否	
				電源 ON 時	工場出荷時	メモリ動作中	スイープ動作中
発生	出力	SBY OPR SUS	出力 OFF（スタンバイ） 出力 ON（オペレート） 出力 ON（サスペンド）	●		○	○
		SBY? OPR? SUS?	出力状態のクエリ 応答 : SBY, OPR または SUS			○	○
	サスペンド HiZ/LoZ	SUZ0 SUZ1	HiZ: 高抵抗出力状態 LoZ: 低抵抗出力状態		●	×	△
		SUZ?	サスペンド HiZ/LoZ のクエリ 応答 : SUZ0 ~ SUZ1			○	○
	発生モード	MD0 MD2 MD4	DC 発生モード スイープ発生モード メモリ・リコール・モード OPR/SUS 中に変更すると SBY になります		●	×	△
		MD?	応答 : MD0, MD2, MD4			○	○

分類	項目	コマンド	内容	初期値		動作可否	
				電源ON時	工場出荷時	メモリ動作中	スイープ動作中
発生	発生ファンクション	VF IF TF（6156のみ）	電圧発生 電流発生 熱起電力発生 OPR/SUS中に電圧⇔電流、熱起電力⇔電流を変更するとSBYになります。		●	×	△
		VF? IF? TF?(6156のみ)	ファンクション／レンジのクエリ 応答 : SVR2 ~ SVR5、SIR1 ~ SIR3、TC1 ~ TC8			○	○
	ファンクション／レンジ	SVR2 SVR3 SVR4 SVR5	電圧発生 30mV 電圧発生 300mV 電圧発生 3V 電圧発生 30V		●	×	×
		SVR?	電圧発生レンジのクエリ 応答 : SVR2 ~ SVR5			○	○
		SVRX0 SVRX1	電圧発生値設定オート・レンジ OFF 電圧発生値設定オート・レンジ ON		●	×	×
		SVRX?	電圧発生値設定オート・レンジのクエリ 応答 : SVRX0 ~ SVRX1				
		SIR1 SIR2 SIR3	電流発生 3mA 電流発生 30mA 電流発生 200mA		●	×	×
		SIR?	電流発生レンジのクエリ 応答 : SIR1 ~ SIR3			○	○
		SIRX0 SIRX1	電流発生値設定オート・レンジ OFF 電流発生値設定オート・レンジ ON		●	×	×
		SIRX?	電流発生値設定オート・レンジのクエリ 応答 : SIRX0 ~ SIRX1				
	（6156のみ）	TC1 TC2 TC3 TC4 TC5 TC6 TC7 TC8	Tタイプ熱電対　熱起電力出力 (CC) Jタイプ熱電対　熱起電力出力 (IC) Eタイプ熱電対　熱起電力出力 (CRC) Kタイプ熱電対　熱起電力出力 (CA) Sタイプ熱電対　熱起電力出力 (PR10) Rタイプ熱電対　熱起電力出力 (PR13) Bタイプ熱電対　熱起電力出力 (PR30) Nタイプ熱電対　熱起電力出力 (N) TC8（Nタイプ）を指定した場合はJSV1（JIS C1602-1981）が選択されていても内部ではJSV0として扱われます。		●	×	×
		TC?	熱電対タイプのクエリ 応答 : TC1 ~ TC7			○	○

分類	項目	コマンド	内容	初期値		動作可否	
				電源ON時	工場出荷時	メモリ動作中	スイープ動作中
発生	熱電対規格の選択（6156のみ）	JSV0 JSV1	熱電対 JIS 規格 C1602-1995(新 JIS) 熱電対 JIS 規格 C1602-1981(旧 JIS) TC8（N タイプ熱電対）が選択されている場合はJSV1 を指定しても内部では JSV0 として扱われます。		●	×	×
		JSV?	熱電対規格のクエリ 応答 : JSV0 ～ JSV1			○	○
	電圧発生値	SOV ±data [unit]	電圧発生値の設定 *1 固定レンジの場合 レンジ固定 :SOV±ddddd レンジ自動 :SOV±dddddunit オートレンジの場合 レンジ自動 :SOV±ddddd レンジ自動 :SOV±dddddunit unit V: ボルト MV: ミリボルト UV: マイクロボルト -1.5mV を設定の例 SOV-1.5E-3 SOV-1.5MV • 各レンジの分解能以下の値は四捨五入されます • 固定レンジでレンジ範囲外の設定はできません • 熱起電力出力時は設定できません		0	×	×
		SOV?	電圧発生値のクエリ *2 応答 : SOV±d.ddddE±d			○	○
	電流発生値	SOI ±data [unit]	電流発生値の設定 *1 固定レンジの場合 レンジ固定 :SOI±ddddd レンジ自動 :SOI±dddddunit オートレンジの場合 レンジ自動 :SOI±ddddd レンジ自動 :SOI±dddddunit unit A: アンペア MA: ミリアンペア UA: マイクロアンペア -1.5mA を設定の例 SOI-1.5E-3 SOI-1.5MA • 各レンジの分解能以下の値は四捨五入されます • 固定レンジでレンジ範囲外の設定はできません • 熱起電力出力時は設定できません		0	×	×
		SOI?	電流発生値のクエリ *2 応答 : SOI±d.ddddE±d			○	○
	熱起電力発生値（6156のみ）	TMP ±data	熱起電力発生値の設定 *1		23	×	×
		TMP?	熱起電力発生値のクエリ 応答 : TMP±dddd.dE±d			○	○

*1: 発生値の設定範囲は機種、レンジ、ファンクションによって異なります。「8. 性能諸元」を参照してください。
*2: 応答の桁数、小数点位置は機種、レンジ、ファンクションによって異なります。「5.7 発生値のデータ出力形式（トーカ・フォーマット）」を参照してください。

分類	項目	コマンド	内容	初期値		動作可否	
				電源ON 時	工場出荷時	メモリ動作中	スイープ動作中
発生	室温補償（6156 のみ）	TCO0 TCO1	室温補償 OFF 室温補償 ON		● 	×	×
		TCO?	室温補償のクエリ 応答 : TCO0 ~ TCO1			○	○
		TMC ±data	室温の設定 data: -25.0 ~ +85.0（単位 :°C）		0	×	×
		TMC?	室温設定値のクエリ 応答 : TMC±dddd.dE±d			○	○
	リミット値	LMV data1[,data2]	電圧リミット値の設定 • 設定範囲 : ±1.0 ~ ±32.0 • data2 を省略した場合 data1 と同じ値の逆極性のリミット値が設定されます • data1 と data2 は逆の極性の値を記述してください • マイナス極性の data が LO 側のリミット値、プラス極性の data が HI 側のリミット値として設定されます		±32V	×	△
		LMV?	電圧リミット値のクエリ 応答 : LMV<-d.ddE±d>,<+d.ddE±d>			○	○
		LMI data1[,data2]	電流リミット値の設定 • 設定範囲 : ±0.005 ~ ±0.220 • data2 を省略した場合 data1 と同じ値の逆極性のリミット値が設定されます • data1 と data2 は逆の極性の値を記述してください • マイナス極性の data が LO 側のリミット値、プラス極性の data が HI 側のリミット値として設定されます		±220 mA	×	△
		LMI?	電流リミット値のクエリ 応答 : LMI<-d.ddE±d>,<+d.ddE±d>			○	○
	リモート・センス	RS0 RS1	2 Wire 4 Wire		●	×	△
		RS?	リモート・センスのクエリ 応答 : RS0 ~ RS1			○	○
	レンジ変更時のカウント・フィクス	CFIX0 CFIX1	レンジ変更時にカウント値を固定しない レンジ変更時にカウント値を固定する		●	×	×
		CFIX?	レンジ変更時のカウント維持のクエリ 応答 : CFIX0 ~ CFIX1			○	○

分類	項目	コマンド	内容	初期値		動作可否	
				電源ON時	工場出荷時	メモリ動作中	スイープ動作中
発生	連続可変桁指定	CUP0	E+0 桁の連続増加			×	×
		CUP1	E+1 桁の連続増加				
		CUP2	E+2 桁の連続増加				
		CUP3	E+3 桁の連続増加				
		CUP4	E+4 桁の連続増加				
		CUP5	E+5 桁の連続増加（6156 のみ）（VS/IS 時のみ）				
		CDN0	E+0 桁の連続減少			×	×
		CDN1	E+1 桁の連続減少				
		CDN2	E+2 桁の連続減少				
		CDN3	E+3 桁の連続減少				
		CDN4	E+4 桁の連続減少				
		CDN5	E+5 桁の連続増加（6156 のみ）（VS/IS 時のみ）				
	極性モード	PLM0	極性モードをバイポーラにする		●	×	×
		PLM1	極性モードをユニポーラにする				
		PLM?	極性モードのクエリ 応答 : PLM0 ~ PLM1			○	○
	スタンバイ・モード	BY0	スタンバイ・モード 0 にする		●	×	×
		BY1	スタンバイ・モード 1 にする				
		BY?	スタンバイ・モードのクエリ 応答 : BY0 ~ BY1			○	○
	発生値のクエリ	SRC?	現在の電圧または電流発生値を応答する *2 （熱起電力の場合も電圧相当値を応答する） 応答 : SOV±d.ddddE±d または SOI±d.ddddE±d			○	○

*1: 発生値の設定範囲は機種、レンジ、ファンクションによって異なります。「8. 性能諸元」を参照してください。

*2: 応答の桁数、小数点位置は機種、レンジ、ファンクションによって異なります。「5.7 発生値のデータ出力形式（トーカ・フォーマット）」を参照してください。

分類	項目	コマンド	内容	初期値		動作可否	
				電源 ON 時	工場 出荷時	メモリ動 作中	スイープ 動作中
メモリ	メモリ設定	Nnnn P	◆ 電圧／電流発生の場合 *1 1. Nnnn,: メモリ番号 0 ～ 4999 2. SVRn, または SIRn, : 発生レンジ 省略時は自動レンジ設定 3. SOV±data, または SOI±data, : 電圧／電流発生値 4. Nnnn: 次のメモリ番号 省略時は自動的にインクリメント 5. SOV±data, または SIR±data, : 次の発生値 ～～ 6. P: メモリ設定モード終了 ◆ 熱起電力発生の場合 1. Nnnn,: メモリ番号 0 ～ 4999 2. TCn, 熱電対タイプ 省略時は T タイプ熱電対に設定される 3. JSVn, : 熱電対規格の選択 省略時は C1602-1995 に設定される 4. TMP±data, : 熱起電力発生温度 5. Nnnn: 次のメモリ番号 省略時は自動的にインクリメント 6. TMP±data, : 次の発生値 ～～ 7. P: メモリ設定モード終了 • 電圧／電流／熱起電力は混在可能です	N0	30mV レンジ 0 *5	×	×
		N?nnn	指定したメモリ番号の発生値のクエリ *2 応答 : N<nnn>,SVR<n>,SOV<±data>,P または N<nnn>,SIR<n>,SOI<±data>,P または N<nnn>,TC<n>,JSV<n>,TMP<±data>,P			○	○
		NP?	メモリ設定状態のクエリ 応答 : 0 メモリ設定終了 1 メモリ設定中	0		○	○
		RSAV	メモリ・データのセーブ実行			×	×
		RLOD	メモリ・データのロード実行			×	×
		RCLR	メモリ・データの初期化実行 • スタンバイになります • セーブされたデータは初期化されません			×	×

*1: 発生値の設定範囲は機種、レンジ、ファンクションによって異なります。「8. 性能諸元」を参照してください。

*2: 応答の桁数、小数点位置は機種、レンジ、ファンクションによって異なります。「5.7 発生値のデータ出力形式（トーカ・フォーマット）」を参照してください。

*5: RINI, *RST コマンドで初期化されません。

分類	項目	コマンド	内容	初期値		動作可否	
				電源ON時	工場出荷時	メモリ動作中	スイープ動作中
メモリ	ファースト・ナンバ ラスト・ナンバ	SC[mmm,]nnn	ファースト No. とラスト No. の設定 SCmmm,nnn mmm: ファースト No. 0 ~ 4999 nnn: ラスト No. 0 ~ 4999 mmm > nnn とすると設定できません • ファースト No. を省略すると 0 に設定されます		0, 4999	×	×
		SC?	ファースト／ラスト No. のクエリ 応答 : SC<mmm>,<nnn> mmm: ファースト No. 0 ~ 4999 nnn: ラスト No. 0 ~ 4999			○	○
	ステップ時間	SPdd.dd	ステップ時間の設定 SPdd.d　dd.d: 0.02 ~ 10.00（単位 sec）		0.1	×	△
		SP?	ステップ時間のクエリ 応答 : SPdd.dd			○	○
	スキャン・モード	SCM0	ホールド・スキャン・モード		●	×	△
		SCM1	シングル・スキャン・モード				
		SCM2	リピート・スキャン・モード				
		SCM?	スキャン・モードのクエリ 応答 : SCM0 ~ SCM2			○	○
	メモリ・ナンバ指定	RCLMnnn	出力するメモリ No. を指定する RCLMnnn,: メモリ No. 0 ~ 4999	0		×	×
		RCLM?	現在のメモリ No. のクエリ 応答 : RCLM000 ~ RCLM4999			○	○
	スキャン開始	*TRG	ホールド・スキャン時 : メモリ No. をインクリメントする シングル／リピート・スキャン時 : スキャンを開始するまたは一時停止する			○	○
	スキャン停止	SWSP	スキャンの停止			○	○
スイープ	スイープ値	SN ±start,	スイープ・データの設定　*1 start: スタート値		0.001mV 0.0001mA 23.1°C	×	×
		±stop,	±stop: ストップ値		0.1mV 0.01mA 33.0°C		
		±step	±step: ステップ値（極性は無視されます）		0.001mV 0.0001mA 0.1°C		
			• スイープ実行時に各値の発生可能な最適な固定レンジになります • 各値の省略はできません • 現在のファンクションの値の設定となります				
		SN?	リニア・スイープ・データのクエリ　*2 応答 : SN<±start>,<±stop>,<±step>			○	○

*1:　発生値の設定範囲は機種、レンジ、ファンクションによって異なります。「8. 性能諸元」を参照してください。
*2:　応答の桁数、小数点位置は機種、レンジ、ファンクションによって異なります。「5.7 発生値のデータ出力形式（トーカ・フォーマット）」を参照してください。

5.8.3 リモート・コマンド一覧

分類	項目	コマンド	内容	初期値		動作可否	
				電源 ON 時	工場 出荷時	メモリ動 作中	スイープ 動作中
スイープ	バイアス値	SB ±data	スイープ・バイアス値の設定 *1 • 現在のファンクションの値の設定となります		0mV 0mA 23.0°C	×	×
		SB?	スイープ・バイアス値のクエリ *2 応答 : SB±data			○	○
	RTB (Return To Bias)	RB0 RB1	OFF（スイープ・ストップ時、最終出力のままとなる） ON（スイープ・ストップ時、バイアス出力に戻る）	●		×	△
		RB?	RTB のクエリ 応答 : RB0 または RB1			○	○
	ステップ時間	SPdd.dd	ステップ時間の設定 *3 SPdd.d dd.d: 0.02 ~ 10.00（単位 s）		0.1	×	△
		SP?	ステップ時間のクエリ *3 応答 : SPdd.dd			○	○
	スイープ・モード	SCM0 SCM1 SCM2	ホールド・スイープ・モード *3 シングル・スイープ・モード リピート・スイープ・モード	●		×	△
		SCM?	スイープ・モードのクエリ *3 応答 : SCM0 ~ SCM2			○	○
	スイープ開始	*TRG	ホールド・スイープ時 : *3 スイープを 1 ステップ進める シングル／リピート・スイープ時 : スイープを開始する、または一時停止する			○	○
	スイープ停止	SWSP	スイープの停止 *3			○	○

*1: 発生値の設定範囲は機種、レンジ、ファンクションによって異なります。「8. 性能諸元」を参照してください。
*2: 応答の桁数、小数点位置は機種、レンジ、ファンクションによって異なります。「5.7 発生値のデータ出力形式（トーカ・フォーマット）」を参照してください。
*3: コマンド一覧の分類 : メモリの項目に記述されているコマンドと同一のコマンドですが、説明上、重複して記述しています。

分類	項目	コマンド	内容	初期値		動作可否	
				電源 ON 時	工場出荷時	メモリ動作中	スイープ動作中
リモート	ブロック・デリミタ	DL0	CR/LF +EOI	●	*4	×	△
		DL1	LF				
		DL2	EOI				
		DL3	LF +EOI • EOI は GPIB の機能のため USB では出力しません				
		DL?	ブロック・デリミタのクエリ 応答 : DL0 ~ DL3			○	○
	サービス・リクエスト	S0	SRQ 発信を許可する		*4	×	△
		S1	SRQ 発信を許可しない	●			
		S?	SRQ 発信許可のクエリ 応答 : S0 ~ S1			○	○
	ステータス	*STB?	ステータス・バイトの読み出し 応答： ddd			○	○
		*SRE *6	サービス・リクエスト・イネーブル・レジスタの設定 (0 ~ 255)	*PSC による	0 *5	○	○
		*SRE? *6	応答： ddd			○	○
		*ESR? *6	スタンダード・イベント・ステータス・レジスタ（ESR）のクエリ 応答： ddd			○	○
		*ESE *6	スタンダード・イベント・ステータス・イネーブル・レジスタの設定 (0 ~ 255)	*PSC による	0 *5	○	○
		*ESE? *6	応答： ddd			○	○
		DSR? *6	デバイス・イベント・ステータス・レジスタ (DSR) のクエリ 応答： ddddd			○	○
		DSE *6	デバイス・イベント・ステータス・イネーブル・レジスタの設定 (0 ~ 65535)		0 *5	○	○
		DSE? *6	応答： ddddd			○	○
		*CLS *6	ステータスのクリア			○	○
	オペレーション・コンプリート	*OPC	全動作終了後、スタンダード・イベント・ステータス・レジスタの LSB をセット			○	○
		*OPC?	応答： 1（全動作終了後）			○	○
		*WAI	全動作終了を待つ（GPIB のみ）			○	○

*4: RINI コマンドで初期化されません。
*5: RINI, *RST コマンドで初期化されません。
*6: 6144 互換モードでは使用できません。

<table>
<tr><th rowspan="2">分類</th><th rowspan="2">項目</th><th rowspan="2">コマンド</th><th rowspan="2">内容</th><th colspan="2">初期値</th><th colspan="2">動作可否</th></tr>
<tr><th>電源
ON 時</th><th>工場
出荷時</th><th>メモリ動
作中</th><th>スイープ
動作中</th></tr>
<tr><td rowspan="2">リモート</td><td rowspan="2">電源 ON 時ステータス・クリア・フラグ</td><td>*PSC<n></td><td>n: ±32767
n≠0 の場合、電源 ON 時に以下のレジスタが初期化されます
n=0 の場合、電源 ON 時に以下のレジスタが初期化されません
• サービス・リクエスト・イネーブル・レジスタ
• スタンダード・イベント・ステータス・イネーブル・レジスタ</td><td></td><td>0</td><td>○</td><td>○</td></tr>
<tr><td>*PSC?</td><td>応答 : 0 または 1（0 以外を設定した場合）</td><td></td><td></td><td>○</td><td>○</td></tr>
<tr><td rowspan="9">システム</td><td>初期化</td><td>C
*RST</td><td>デバイス・クリア
パラメータの初期化</td><td></td><td></td><td>○
○</td><td>○
○</td></tr>
<tr><td>機器情報</td><td>*IDN?</td><td>機器の問い合わせ
応答 : ADC Corp.6146,nnnnnnnnn,xxxx
または
ADC Corp.6156,nnnnnnnnn,xxxx
nnnnnnnnn: シリアル番号
xxxx: ソフトウェア・レビジョン</td><td></td><td></td><td>○</td><td>○</td></tr>
<tr><td>セルフ・テスト</td><td>*TST?</td><td>実行および結果読み出し
応答 : 0; Pass
1; Fail
（オペレート中は実行できません）</td><td></td><td></td><td>×</td><td>×</td></tr>
<tr><td>エラー・ログ</td><td>ERR?</td><td>エラー・ログ内容の読み出し
応答 : ± ddd, "xxxxxxxxxx"
• エラー・ログは FIFO 方式で最大 20 個記録されます
• 20 を超えるエラーが発生した場合は最後に保存されたエラーが -350,"Queue overflow" と置き換わります
• エラーがない場合は +000,"No error" を応答します</td><td></td><td></td><td>○</td><td>○</td></tr>
<tr><td rowspan="2">通知ブザー</td><td>NZ0
NZ1</td><td>OFF
ON</td><td></td><td>●</td><td>×</td><td>△</td></tr>
<tr><td>NZ?</td><td>通知ブザー設定のクエリ
応答 : NZ0 ～ NZ1</td><td></td><td></td><td>○</td><td>○</td></tr>
<tr><td rowspan="2">リミット検出ブザー</td><td>UZ0
UZ1</td><td>OFF
ON</td><td></td><td>●</td><td>×</td><td>△</td></tr>
<tr><td>UZ?</td><td>リミット検出ブザー設定のクエリ
応答 : UZ0 ～ UZ1</td><td></td><td></td><td>○</td><td>○</td></tr>
</table>

分類	項目	コマンド	内容	初期値		動作可否	
				電源 ON 時	工場出荷時	メモリ動作中	スイープ動作中
システム	インタロック設定	OP0 OP1 OP2 OP3 OP4	STBY IN 信号入力 OPR/STBY IN 信号入力 INTERLOCK IN 信号入力 OPR OUT 信号出力 OPR/SUS IN 信号入力 （OPn コマンドによりスタンバイになります）		●	×	×
		OP?	インタロック設定のクエリ 応答 : OP0 ～ OP4			○	○
	同期信号設定	CRP0 CRP1	READY OUT 信号出力 SYNC OUT 信号出力		●	×	△
		CRP?	同期信号設定のクエリ 応答 : CRP0 ～ CRP1			○	○
	READY 信号出力遅延時間	ROD0 ROD1	READY OUT 信号出力遅延時間 10ms READY OUT 信号出力遅延時間 50ms		●	×	△
		ROD?	READY OUT 信号出力遅延時間のクエリ 応答 : ROD0 ~ ROD1			○	○
	オプション	*OPT?	オプションのクエリ 応答 : オプションなし : 0 オプションあり : "BCD I/F"			○	○
	オペレート・ホールドのクエリ	OPH?	オペレート・ホールド機能のクエリ 応答 : 0 OFF 1 ON			○	○
	6144 互換モード	CMP0 CMP1	互換モード OFF 互換モード ON		●	×	×
		CMP?	互換モードのクエリ 応答 : CMP0 ~ CMP1			○	○
	パラメータ・セーブ	*SAV0 *SAV1 *SAV2 *SAV3	設定されているパラメータを、不揮発メモリの領域「0」にセーブ 設定されているパラメータを、不揮発メモリの領域「1」にセーブ 設定されているパラメータを、不揮発メモリの領域「2」にセーブ 設定されているパラメータを、不揮発メモリの領域「3」にセーブ			×	△
		SINI	工場出荷時の値を、「0」～「3」の領域すべてに設定			×	△

5.8.3 リモート・コマンド一覧

<table>
<tr><th rowspan="2">分類</th><th rowspan="2">項目</th><th rowspan="2">コマンド</th><th rowspan="2">内容</th><th colspan="2">初期値</th><th colspan="2">動作可否</th></tr>
<tr><th>電源
ON 時</th><th>工場
出荷時</th><th>メモリ動
作中</th><th>スイープ
動作中</th></tr>
<tr><td rowspan="2">システム</td><td rowspan="2">パラメータ・ロード</td><td>*RCL0
*RCL1
*RCL2
*RCL3</td><td>不揮発メモリの領域「0」のデータを、設定パラメータとしてロード
不揮発メモリの領域「1」のデータを、設定パラメータとしてロード
不揮発メモリの領域「2」のデータを、設定パラメータとしてロード
不揮発メモリの領域「3」のデータを、設定パラメータとしてロード</td><td></td><td></td><td>×</td><td>×</td></tr>
<tr><td>RINI</td><td>工場出荷時の値を、設定パラメータとしてロード</td><td></td><td></td><td>×</td><td>×</td></tr>
<tr><td rowspan="6">校正</td><td rowspan="2">校正 SW</td><td>CAL0
CAL1</td><td>OFF（校正モードから抜ける）
ON （校正モードに入る）</td><td>●</td><td></td><td>×</td><td>×</td></tr>
<tr><td>CAL?</td><td>校正 SW のクエリ
応答 : CAL0 ~ CAL1</td><td></td><td></td><td>○</td><td>○</td></tr>
<tr><td rowspan="2">校正ファンクション</td><td>XF0
XF1
XF2
XF3
XF4</td><td>電圧発生校正
電流発生校正
電圧リミッタ校正
電流リミッタ校正
DAC 校正（6156 のみ）</td><td></td><td></td><td>×</td><td>×</td></tr>
<tr><td>XF?</td><td>校正ファンクション設定のクエリ
応答 :XF0 ~ XF4</td><td></td><td></td><td>×</td><td>×</td></tr>
<tr><td rowspan="2">校正レンジ</td><td>XRn</td><td>校正レンジの設定
<table>
<tr><th></th><th>電圧レンジ</th><th>電流レンジ</th></tr>
<tr><td>XR1</td><td>-</td><td>3mA レンジ</td></tr>
<tr><td>XR2</td><td>30mV レンジ</td><td>30mA レンジ</td></tr>
<tr><td>XR3</td><td>300mV レンジ</td><td>200mA レンジ
（IL レンジ応答）</td></tr>
<tr><td>XR4</td><td>3V レンジ</td><td>-</td></tr>
<tr><td>XR5</td><td>30V レンジ
（VL レンジ応答）</td><td>-</td></tr>
</table>
リミッタ校正時はエラーとなります</td><td></td><td></td><td>×</td><td>×</td></tr>
<tr><td>XR?</td><td>校正レンジ設定のクエリ
応答 :XR1 ~ XR5</td><td></td><td></td><td>×</td><td>×</td></tr>
</table>

分類	項目	コマンド	内容	初期値		動作可否	
				電源 ON 時	工場 出荷時	メモリ動 作中	スイープ 動作中
校正	校正実行	XS0 XS1	+ 極性（リミッタ校正時は High 側リミッタ） - 極性（リミッタ校正時は Low 側リミッタ）			×	×
		XS?	± 校正モードのクエリ 応答 :XS0 ~ XS1			×	×
		XN0 XN1	ゼロ校正 フル校正			×	×
		XN?	ゼロ／フル校正モードのクエリ 応答 :XN0 ~ XN1			×	×
		XM0 XM1	通常発生モード DMM 入力モード			×	×
		XM?	校正モードのクエリ 応答 :XM0 ~ XM1			×	×
	校正データの保存 (RAM → ROM)	XWR	校正データを不揮発メモリに保存			×	×
	校正データの初期化 (RAM)	XINI	校正データの初期化 • 不揮発メモリ内の校正データは影響されません			×	×
		XCLR	現在のファンクション、レンジ、極性の校正データを初期化 • 不揮発メモリ内の校正データは影響されません			×	×
	校正データ入力	XDT±ddddE±d	DMM 読み込みデータ入力			×	×
	DAC 校正実行 （6156 のみ）	XK1 XK2 XK3 XK4 XK5 XK6 XK7 XK8 XK9 XK10 XK11 XK12	ym0 校正 (DMM 入力モード) ym1 校正 (DMM 入力モード) ym10 校正 (DMM 入力モード) ym11 校正 (DMM 入力モード) ym12 校正 (DMM 入力モード) ymi1 校正 (DMM 入力モード) ymi2 校正 (DMM 入力モード) ymi3 校正 (DMM 入力モード) ymi4 校正 (DMM 入力モード) ymi5 校正 (DMM 入力モード) ymi6 校正 (DMM 入力モード) ymi7 校正 (DMM 入力モード)			×	×

5.8.4　6144 互換リモート・コマンド一覧

在来機種の 6144 との互換のためのコマンドです。

特に断りのない限り、互換モードにかかわらずすべてのコマンドが使用可能です。

6146/6156 の機能を最大に使用するためにも、特別に互換を重視する場合以外は、通常のコマンドの使用を推奨します。

「5.1.2 互換性について」も参照してください。

分類	項目	コマンド	内容	初期値 電源ON時	初期値 工場出荷時	動作可否 メモリ動作中	動作可否 スイープ動作中
発生	出力	E	出力 ON（オペレート）			○	○
		H	出力 OFF（スタンバイ）	●			
		E? H?	出力状態のクエリ 応答 : オペレート中 : E スタンバイ中 : H サスペンド中 : H			○	○
	ファンクション／レンジ	V2	電圧発生 30mV			×	×
		V3	電圧発生 300mV				
		V4	電圧発生 3V		●		
		V5	電圧発生 30V				
		V6	電圧発生 30V（V5 コマンドと同一の機能です）				
		I1	電流発生 3mA				
		I2	電流発生 30mA				
		I3	電流発生 200mA				
		V? I?	ファンクション／レンジのクエリ 応答 : V2 ～ V5, I1 ～ I3			○	○
	電圧／電流発生値	D±data [unit]	電圧／電流発生値の設定 *1 レンジ固定 :D±ddddd レンジ自動 :D±dddddunit （NRF 可） V: ボルト MV: ミリボルト UV: マイクロボルト MA: ミリアンペア UA: マイクロアンペア -1.5mV を設定の例 D-1.5（あらかじめ V2 または V3 に設定する） +1.5V を設定の例 D1.5（あらかじめ V4, V5 または V6 に設定する） D1.5V • 熱起電力出力時は設定できません • このコマンドの実行はオート・レンジの OFF/ON(SVRXn/SIRXn) には影響されません		0	×	×
		D?	発生値のクエリ *2 応答 : DV±ddddE±d DI±ddddE±d			○	○

*1:　発生値の設定範囲は機種、レンジ、ファンクションによって異なります。「8. 性能諸元」を参照してください。

*2:　応答の桁数、小数点位置は機種、レンジ、ファンクションによって異なります。「5.7 発生値のデータ出力形式（トーカ・フォーマット）」を参照してください。

分類	項目	コマンド	内容	初期値 電源ON時	初期値 工場出荷時	動作可否 メモリ動作中	動作可否 スイープ動作中
連続可変	連続可変桁指定	K0	E+0 桁の連続増加			×	×
		K1	E+1 桁の連続増加				
		K2	E+2 桁の連続増加				
		K3	E+3 桁の連続増加				
		K4	E+0 桁の連続減少				
		K5	E+1 桁の連続減少				
		K6	E+2 桁の連続減少				
		K7	E+3 桁の連続減少				
メモリ	メモリ設定	Nnnn C3	1. Nnnn,: メモリ・チャンネル　0 ~ 159 2. D±ddddd または D±dddddunit : 電圧／電流発生値（V/I 混在可） 3. Vn または In : 発生レンジ（オート・レンジのときは省略可） 4. Nnnn: 次のチャンネル 省略時は自動的にインクリメント 5. D±ddddd または D±dddddunit : 次の発生値 ~ ~ 6. C3: メモリ設定モード終了 • 電圧／電流／熱起電力は混在可能です • 6144 互換モード時に使用可能なメモリ番号は 0 ~ 159 となります	N0	30mV レンジ 0	×	×
		N?	現在の設定メモリ・チャンネルのクエリ 応答 : N000 ~ N159	0		○	○
		P?	メモリ設定モードのクエリ 応答 : P0 メモリ設定モード OFF P1 メモリ設定モード ON	0		○	○
	ファースト・チャンネル ラスト・チャンネル	SC[mmm,]nnn	ファースト No. とラスト No. の設定 SCmmm,nnn mmm: ファースト No. 0 ~ 159 nnn: ラスト No. 0 ~ 159 • mmm > nnn の設定できません • ファースト No. を省略すると 0 に設定されます • 6144 互換モード時に使用可能なメモリ番号は 0 ~ 159 となります		0, 159 互換モード時の初期値	×	△
		SC?	ファースト／ラスト・チャンネルのクエリ 応答 : SCmmm nnn mmm: ファースト・チャンネル 000 ~ 159 nnn: ラスト・チャンネル 000 ~ 159 • 互換モードでの応答です			○	○
	ステップ時間	SIddd	スキャン・ステップ時間の設定 SIddd　　ddd: 1 ~ 100（単位 ×E-1s）		1 互換モード時の初期値	×	△
		SI?	ステップ時間のクエリ 応答 : SIddd 0.1s 未満の設定に対する応答は出力されません			○	○

5.8.4 6144 互換リモート・コマンド一覧

分類	項目	コマンド	内容	初期値 電源ON時	初期値 工場出荷時	動作可否 メモリ動作中	動作可否 スイープ動作中
メモリ	スキャン開始 *7 *8	T1	ホールド・スキャンのメモリ番号をインクリメントする			×	×
		T2	シングル・スキャンの開始			×	×
		T3	リピート・スキャンの開始			×	×
		T?	現在実行中の（最後に実行した）スキャン・モードのクエリ 応答 : T2 シングル・スキャン T3 リピート・スキャン（初期値）			○	○
	スキャン停止 *7 *8	C1	スキャンの中止			○	×
		C2	スキャンの一時停止 T1, T2, T3 で再スタート			○	×
	設定データのバッファ	B	このコマンドに続けて、発生値、レンジ設定コマンドを記述するとその値がバッファへ格納され、E/OPR/GET コマンドにより出力されます。 1 度出力されるとバッファは消去されます。 熱起電力出力データはバッファに格納されません。バッファに格納されるのは Vn/In/Ddddd コマンドのみです。			×	×
		B?	バッファに格納のクエリ 応答 : B0 格納されていない B1 格納されている	B0		○	○
システム	初期化	C	設定の初期化 • メモリ内データは初期化されません • 互換モードでは機能が異なります			○	○
		C0					
	外部スイッチのクエリ	O?	オペレート・ホールド・モードのクエリ 応答 : O0 OFF O1 ON			○	○
		X?	EXT.CAL スイッチのクエリ 応答 : X0 OFF X1 ON			○	○

*7: 6144 互換モードでのみ使用可能です。
*8: スイープ発生モードでは使用できません。

5.9 サンプル・プログラム

ここでは、本器をコンピュータから操作する基本的なプログラム例を説明します。

5.9.1 インタフェース制御部分の共通サブプロシージャ

5.9.2 プログラム例 1- 直流電圧発生、DMM での電圧・電流の測定

5.9.3 プログラム例 2- メモリ設定・スキャン動作、DMM での電圧・電流の測定

5.9.4 プログラム例 3- スイープ動作、DMM での電圧・電流の測定

- DMM (Digital Multimeter) : ここでは弊社の 7351 A/E を使用します。
  *7351E では使用できるインタフェースは USB のみとなります。
- プログラム・コードおよび解説コメントでは本器を VIG、デバイス・イベント・ステータス・レジスタ (Device Event Status Register) を DSR、ステータス・バイト・レジスタ (Status Byte Register) を STB と表記します。
- インタフェースの制御部分は共通のサブプロシージャになっており、
  「5.9.1 インタフェース制御部分の共通サブプロシージャ」にまとめてあります。

【動作確認環境】

使用コンピュータ :	富士通株式会社　FMV-D3220
OS:	Windows（R）XP Professional
CPU:	Intel（R）Pentium（R）4 524（3.06GHz）
使用言語 :	Microsoft Office Excel Visual Basic for Applications

インタフェースが GPIB の場合

GPIB ハードウェア :	National Instruments 社製 NI GPIB-USB-HS
使用モジュール :	Niglobal.bas, Vbib-32.bas （NI GPIB-USB-HS 付属 NI-488.2 ドライバソフトウェア インタフェースファイル）

本器の GPIB アドレス : 1
DMM の GPIB アドレス : 2

インタフェースが USB の場合

使用モジュール :	ausb.bas （弊社製 ADC 計測器 USB ドライバ 付属インタフェース ソフトウェア）
本器の USB.ID:	1
DMM の USB.ID:	2

5.9.1 インタフェース制御部分の共通サブプロシージャ

5.9.1.1 GPIB でのプログラム例で共通のプロシージャ

```
' GPIB データ送信 サブプロシージャ
Sub gpibwrt(intDes As Integer, strCmd As String)

  Call ibwrt(intDes, strCmd & vbLf)              ' LF 文字を追加して送信

End Sub


' GPIB データ受信 サブプロシージャ
Sub gpibrd(intDes As Integer, strData As String)

  Dim strRawStr As String * 30                   ' データ受信用バッファ
  Dim intSize As Integer                         ' 取出す文字数

  Call ibrd(intDes, strRawStr)                   ' 受信 ( バッファ数のみ )

  If InStr(strRawStr, vbCr) <> 0 Then            ' CR または LF 文字の検索
      intSize = InStr(strRawStr, vbCr) - 1
    ElseIf InStr(strRawStr, vbLf) <> 0 Then
      intSize = InStr(strRawStr, vbLf) - 1
  End If

  If intSize = 0 Then                            ' 検索結果確認
      strData = ""
    Else
      strData = Left$(strRawStr, intSize)        ' CR/LF 文字を除いて取り出す
  End If

End Sub
```

5.9.1.2 USB でのプログラム例で共通のプロシージャ

```
' USB ドライバの初期化 サブプロシージャ
Sub aUSBSta(lngTimeOut As Long)

  Dim lngRet As Long                             ' USB ドライバの戻り値

  lngRet = ausb_start(lngTimeOut)
  If lngRet = OK Then
      blnUSBComErr = False                       ' OK の場合
    Else
      blnUSBComErr = True                        ' NG の場合

      MsgBox"USB 初期化エラー ", vbCritical, " エラー :" & lngRet
  End If

  Call mSecSleep(100)                            ' USB 初期化待機 (100ms)
```

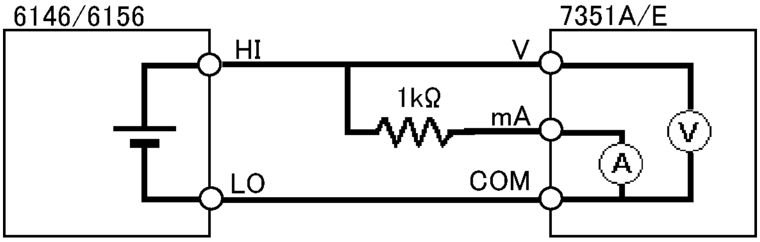
```
End Sub


' USB ドライバの停止 サブプロシージャ
Sub aUSBEnd()

  Dim lngRet As Long                             ' USB ドライバの戻り値

  lngRet = ausb_end()                            ' USB 終了
  If lngRet <> OK Then                           ' NG の場合
    MsgBox "USB 終了エラー ", vbExclamation
  End If

End Sub


' デバイス（機器）のオープンとハンドル取得 サブプロシージャ
Sub aUSBOpn(lngHdl As Long, lngUSBID As Long)

  Dim lngRet As Long                             ' USB ドライバの戻り値

  lngRet = ausb_open(lngHdl, lngUSBID)           ' デバイスをオープン
  If lngRet = OK Then
      blnUSBComErr = False                       ' OK の場合
    Else
      blnUSBComErr = True                        ' NG の場合

      MsgBox " デバイスオープンエラー ", vbCritical, "ID：" & lngUSBID
  End If

End Sub


' デバイス（機器）のクローズ サブプロシージャ
Sub aUSBCls(lngHdl As Long)

  Dim lngRet As Long                             ' USB ドライバの戻り値

  lngRet = ausb_close(lngHdl)                    ' デバイスをクローズ
  If lngRet <> OK Then                           ' NG の場合
      MsgBox " デバイスクローズエラー ", vbCritical
  End If

End Sub


' USB ドライバ データ送信 サブプロシージャ
Sub aUSBWrt(lngHdl As Long, strCmd As String)

  Dim lngRet As Long                             ' USB ドライバの戻り値

  lngRet = ausb_write(lngHdl, strCmd)            ' 送信
  Call mSecSleep(10)                             ' USB 送信処理待機
  If lngRet = OK Then
      blnUSBComErr = False                       ' OK の場合
    Else
      blnUSBComErr = True                        ' NG の場合
      MsgBox strCmd, vbCritical, " 送信エラー "
```

```
  End If

End Sub


' USB ドライバ データ受信 サブプロシージャ
Sub aUSBRd(lngHdl As Long, strData As String)

  Dim lngRet As Long
  Dim lngSize As Long                                 ' 受信有効文字数
  Dim strRawStr As String

  lngRet = ausb_read(lngHdl, strRawStr, 50, lngSize)
                                                      ' データ受信
  If lngRet = OK Then                                 ' OK の場合
      strData = Left$(strRawStr, lngSize - 1)         ' 有効文字数を取り出す
      strData = Replace(strData, vbCr, "")            ' CR 文字を削除
      strData = Replace(strData, vbLf, "")            ' LF 文字を削除
      blnUSBComErr = False
    Else                                              ' NG の場合
      lngRet = ausb_clear(lngHdl)                     ' デバイス・クリア
      blnUSBComErr = True
      MsgBox "受信エラー", vbCritical
  End If

End Sub
```

5.9.2 プログラム例 1- 直流電圧発生、DMM での電圧・電流の測定

本器で直流電圧を発生させ、負荷抵抗 (1kΩ) にかかる電圧・電流を DMM で 1 回計測します。計測値は Excel シートのセルに格納します。

接続図



ファンクション : 直流電圧発生
発生電圧 : +5V
センス状態 ：内部（2W）

使用 DMM: 7351A/E

測定ファンクション : 直流電圧、直流電流
測定レンジ : オート
サンプリング・レート : SLOW1（積分時間 : 100ms）
ヘッダ : ON

（実行結果例）

DCV_ +05.0361E+00　　DCI_ +005.034E-03

5.9.2.1 GPIB でのプログラム例

```
Option Explicit                                  ' すべての変数を明示的に宣言

Private Const intIFBoardAdr As Integer = 0       ' GPIB ボードのアドレスを 0
Private Const intVIGAdr As Integer = 1           ' VIG のアドレスを 1
Private Const intDMMAdr As Integer = 2           ' DMM のアドレスを 2
Private Const lngStaRow As Long = 1              ' 測定データを入れるセルの行
Private Const lngVDataCol As Long = 1            ' 測定電圧データを入れる桁
Private Const lngIDataCol As Long = 2            ' 測定電流データを入れる桁

Public Sub DCGenerateAndMeasure_GPIB()

  Dim intVIGDes As Integer                       ' VIG のデバイス・ディスクリプタ
  Dim intDMMDes As Integer                       ' DMM のデバイス・ディスクリプタ
  Dim strRecvStr As String                       ' 受信データ
  Dim lngVIGDSR As Long                          ' VIG の DSR 値
  Dim lngDMMSTB As Long                          ' DMM の STB 値
```

5.9.2 プログラム例 1- 直流電圧発生、DMM での電圧・電流の測定

```
' デバイス(VIG, DMM)を開いて初期化(タイムアウト10s)
Call ibdev(intIFBoardAdr, intVIGAdr, 0, T10s, 1, 0, intVIGDes)
Call ibdev(intIFBoardAdr, intDMMAdr, 0, T10s, 1, 0, intDMMDes)

' 送受信ごとにアドレス設定を行う
Call ibconfig(intVIGDes, IbcUnAddr, 1)
Call ibconfig(intDMMDes, IbcUnAddr, 1)

' デバイス・クリアおよびパラメータの初期化
Call gpibwrt(intVIGDes, "C,*RST")
Call gpibwrt(intDMMDes, "C,*RST")

' VIGの設定
Call gpibwrt(intVIGDes, "VF")                ' 電圧発生ファンクション
Call gpibwrt(intVIGDes, "SOV+5V")            ' 出力電圧を5V（オート・レンジ）

' DMMの設定
Call gpibwrt(intDMMDes, "F1")                ' 電圧測定ファンクション
Call gpibwrt(intDMMDes, "PR3")               ' サンプリング・レートをSLOW1
Call gpibwrt(intDMMDes, "TRS3")              ' トリガをBUS
Call gpibwrt(intDMMDes, "H1")                ' ヘッダON
Call gpibwrt(intDMMDes, "*CLS")              ' ステータスをクリア

' VIG動作
Call gpibwrt(intVIGDes, "OPR")               ' VIGの出力をONにする
Do
  Call gpibwrt(intVIGDes, "DSR?")            ' DSR問い合わせ
  Call gpibrd(intVIGDes, strRecvStr)         ' データ受信
  lngVIGDSR = CLng(strRecvStr)               ' DSR値格納
  lngVIGDSR = lngVIGDSR And 1024             ' READY Out bitとの論理積
Loop Until lngVIGDSR = 1024                  ' READY Out bitが立つまで繰り返し

' DMM動作
Call gpibwrt(intDMMDes, "*TRG")              ' トリガを送信、測定開始
Do
  Call gpibwrt(intDMMDes, "*STB?")           ' STB問い合わせ
  Call gpibrd(intDMMDes, strRecvStr)         ' データ受信
  lngDMMSTB = CLng(strRecvStr)               ' STB値格納
  lngDMMSTB = lngDMMSTB And 16               ' Message Available bitとの論理積
Loop Until lngDMMSTB = 16                    ' 測定終了まで繰り返し

Call gpibrd(intDMMDes, strRecvStr)           ' 測定データを受信
' 測定データを指定セルに代入
ActiveSheet.Cells(lngStaRow, lngVDataCol).value = strRecvStr

Call gpibwrt(intDMMDes, "F5")                ' 電流測定ファンクション

Call gpibwrt(intDMMDes, "*TRG")              ' トリガを送信、測定開始
Do
    Call gpibwrt(intDMMDes, "*STB?")         ' STB問い合わせ
    Call gpibrd(intDMMDes, strRecvStr)       ' データ受信
    lngDMMSTB = CLng(strRecvStr)             ' STB値格納
    lngDMMSTB = lngDMMSTB And 16             ' Message Available bitとの論理積
Loop Until lngDMMSTB = 16                    ' 測定終了まで繰り返し
```

```
  Call gpibrd(intDMMDes, strRecvStr)            ' 測定データを受信
  ' 測定データを指定セルに代入
  ActiveSheet.Cells(lngStaRow, lngIDataCol).value = strRecvStr

  Call gpibwrt(intVIGDes, "SBY")                ' VIG の出力を OFF にする
  Call ibonl(intVIGDes, 0)                      ' VIG をオフラインにする
  Call ibonl(intDMMDes, 0)                      ' DMM をオフラインにする

End Sub
```

5.9.2.2 USB でのプログラム例

```
Option Explicit                                 ' すべての変数を明示的に宣言

Public blnUSBComErr As Boolean                  ' USB 送受信のエラー発生状態
Private Const OK As Integer = 0                 ' 定数「OK」を宣言
Private Const lngVIGID As Long = 1              ' VIG の USB_ID...1
Private Const lngDMMID As Long = 2              ' DMM の USB_ID...2
Private Const lngTimeOut As Long = 10           ' USB タイムアウト時間 [s]

Private Const lngStaRow As Long = 1             ' 測定データを入れるセルの行
Private Const lngVDataCol As Long = 1           ' 測定電圧データを入れる桁
Private Const lngIDataCol As Long = 2           ' 測定電流データを入れる桁


Public Sub DCGenerateAndMeasure_USB()

  Dim lngVIGHdl As Long                         ' VIG の USB ハンドル
  Dim lngDMMHdl As Long                         ' DMM の USB ハンドル
  Dim strRecvStr As String                      ' データ受信用バッファ
  Dim lngVIGDSR As Long                         ' VIG の DSR 値
  Dim lngDMMSTB As Long                         ' DMM の STB 値


  Call aUSBSta(lngTimeOut)                      ' USB ドライバ初期化

  ' デバイス（機器）のオープンとハンドルの取得
  Call aUSBOpn(lngVIGHdl, lngVIGID)             ' VIG をオープン
  Call aUSBOpn(lngDMMHdl, lngDMMID)             ' DMM をオープン
  If blnUSBComErr = True Then                   ' USB エラー確認
    GoTo ErrExit
  End If

  ' デバイス・クリアおよびパラメータの初期化
  Call aUSBWrt(lngVIGHdl, "C,*RST")
  Call aUSBWrt(lngDMMHdl, "C,*RST")

  ' VIG の設定
  Call aUSBWrt(lngVIGHdl, "VF")                 ' 電圧発生ファンクション
  Call aUSBWrt(lngVIGHdl, "SOV+5V")             ' 電圧を 5V（オート・レンジ）
  Call aUSBWrt(lngVIGHdl, "*CLS")               ' ステータスをクリア

  ' DMM の設定
  Call aUSBWrt(lngDMMHdl, "F1")                 ' 電圧測定ファンクション
```

```
Call aUSBWrt(lngDMMHdl, "PR3")                    ' サンプリング・レートを SLOW1
Call aUSBWrt(lngDMMHdl, "TRS3")                   ' トリガを BUS
Call aUSBWrt(lngDMMHdl, "H1")                     ' ヘッダ ON
Call aUSBWrt(lngDMMHdl, "*CLS")                   ' ステータスをクリア

' VIG の設定動作終了を待つ
Do
  Call aUSBWrt(lngVIGHdl, "*OPC?")                ' 全動作終了？
  Call aUSBRd(lngVIGHdl, strRecvStr)              ' データ受信

  ' USB 送受信にエラーないか確認
  If blnUSBComErr = True Then
    GoTo ErrExit
  End If
Loop Until strRecvStr = "1"


' DMM の設定動作終了を待つ
Do
  Call aUSBWrt(lngDMMHdl, "*OPC?")                ' 全動作終了？
  Call aUSBRd(lngDMMHdl, strRecvStr)              ' データ受信

  ' USB 送受信にエラーないか確認
  If blnUSBComErr = True Then
    GoTo ErrExit
  End If
Loop Until strRecvStr = "1"

' VIG 動作
Call aUSBWrt(lngVIGHdl, "OPR")                    ' VIG の出力を ON にする

Do
  Call aUSBWrt(lngVIGHdl, "DSR?")                 ' DSR 問い合わせ
  Call aUSBRd(lngVIGHdl, strRecvStr)              ' データ受信
  lngVIGDSR = CLng(strRecvStr)                    ' DSR 値格納
  lngVIGDSR = lngVIGDSR And 1024                  ' Ready Out bit との論理積
  If blnUSBComErr = True Then                     ' USB 送受信エラー確認
    GoTo ErrExit
  End If
Loop Until lngVIGDSR = 1024                       ' Ready Out bit が立つまで繰り返し

' DMM 動作
Call aUSBWrt(lngDMMHdl, "*TRG")                   ' トリガを送信、測定開始
Do
  Call aUSBWrt(lngDMMHdl, "*STB?")                ' STB 問い合わせ
  Call aUSBRd(lngDMMHdl, strRecvStr)              ' データ受信
  lngDMMSTB = CLng(strRecvStr)                    ' STB 値格納
  lngDMMSTB = lngDMMSTB And 16                    ' Message Available bit との論理積
  If blnUSBComErr = True Then                     ' USB 送受信エラー確認
    GoTo ErrExit
  End If
Loop Until lngDMMSTB = 16                         ' 測定終了まで繰り返し

Call aUSBRd(lngDMMHdl, strRecvStr)                ' 測定データを受信
' 測定データを指定セルに代入
ActiveSheet.Cells(lngStaRow, lngVDataCol).Value = strRecvStr
```

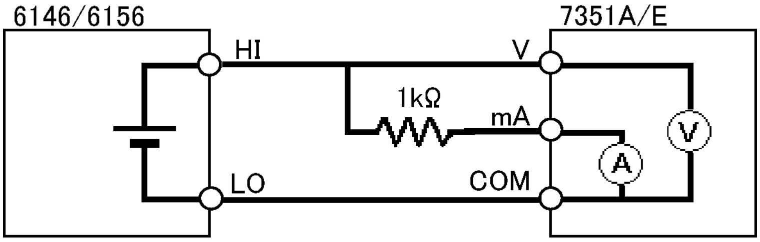
```
  Call aUSBWrt(lngDMMHdl, "F5")                  ' 電流測定ファンクション

  Call aUSBWrt(lngDMMHdl, "*TRG")                ' トリガを送信、測定開始
  Do
    Call aUSBWrt(lngDMMHdl, "*STB?")             ' STB 問い合わせ
    Call aUSBRd(lngDMMHdl, strRecvStr)           ' データ受信
    lngDMMSTB = CLng(strRecvStr)                 ' STB 値格納
    lngDMMSTB = lngDMMSTB And 16                 ' Message Available bit との論理積
    If blnUSBComErr = True Then                  ' USB 送受信エラー確認
      GoTo ErrExit
    End If
  Loop Until lngDMMSTB = 16                      ' 測定終了まで繰り返し

  Call aUSBRd(lngDMMHdl, strRecvStr)             ' 測定データを受信
  ' 測定データを指定セルに代入
  ActiveSheet.Cells(lngStaRow, lngIDataCol).Value = strRecvStr


ErrExit:

  Call aUSBWrt(lngVIGHdl, "SBY")                 ' VIG の出力を OFF にする

  ' デバイス（機器）のクローズ
  Call aUSBCls(lngVIGHdl)                        ' VIG
  Call aUSBCls(lngDMMHdl)                        ' DMM

  ' USB ドライバの終了
  Call aUSBEnd

End Sub
```

5.9.3　プログラム例 2- メモリ設定・スキャン動作、DMM での電圧・電流の測定

本器のメモリに 0V → +5V → 0V の電圧発生を 100mV ステップで設定し、メモリ・リコール・モードで直流電圧を発生させ、DMM で電圧・電流の測定を行います。

このリコール動作と測定動作を繰り返して 2 回行います。
繰り返し数、測定回数、メモリ番号、発生値、測定値を Excel シートのセルに格納します。

接続図



ファンクション :　直流電圧発生
電圧発生範囲 :　0V ~ +5V
ステップ電圧 :　100mV
センス状態 :　内部（2W）
メモリ・スキャン・モード : ステップ

使用 DMM:　7351A/E

測定ファンクション :　直流電圧、直流電流
測定レンジ :　オート
サンプリング・レート :　MED（積分時間 : 20ms）
ヘッダ :　OFF

（実行結果例）

（繰返し数）	（測定数）	（メモリ番号）	（発生電圧値）	（測定電圧値）	（測定電流値）
1	1	0	+00.0000E-3	-000.007E-03	-000.001E-03
1	2	1	+100.000E-3	+091.004E-03	+000.090E-03
1	3	2	+200.000E-3	+182.045E-03	+000.178E-03
.	.	.	.	.	.
省略	.	.	.	.	.
.	.	.	.	.	.
2	200	98	+200.000E-3	+0182.04E-03	-000.181E-03
2	201	99	+100.000E-3	+091.006E-03	+000.090E-03
2	202	100	+00.0000E-3	-000.006E-03	-000.001E-03

5.9.3.1 GPIB でのプログラム例

```
Option Explicit                                    ' すべての変数を明示的に宣言

Private Const intIFBoardAdr As Integer = 0         ' GPIB ボードのアドレスを 0
Private Const intVIGAdr As Integer = 1             ' VIG のアドレスを 1
Private Const intDMMAdr As Integer = 2             ' DMM のアドレスを 2

' 動作設定パラメータ
Private Const dblVMin As Double = 0                ' 開始 ( 最小 ) 電圧 [V]
Private Const dblVMax As Double = 5                ' 折返し ( 最大 ) 電圧 [V]
Private Const dblVStep As Double = 0.1             ' 電圧ステップ値 [V]
Private Const intRepSet As Integer = 2             ' 繰り返し回数

' データ格納するシートの設定
Private Const lngStaRow As Long = 1                ' データを入れる開始行
Private Const lngRepCntCol As Long = 1             ' 繰り返し回数の桁
Private Const lngMeasNoCol As Long = 2             ' 測定回数の桁
Private Const lngMemNoCol As Long = 3              ' メモリ No. の桁
Private Const lngGenerateVCol As Long = 4          ' 発生電圧値の桁
Private Const lngVDataCol As Long = 5              ' 測定電圧の桁
Private Const lngIDataCol As Long = 6              ' 測定電流の桁



Public Sub SetMemAndMeasure_GPIB()
  Dim intVIGDes As Integer                         ' VIG のデバイス・ディスクリプタ
  Dim intDMMDes As Integer                         ' DMM のデバイス・ディスクリプタ
  Dim strRecvStr As String                         ' 受信文字列

  Dim strSendCmd As String                         ' メモリ設定用コマンド文字列
  Dim dblGParam As Double                          ' 発生値パラメータ
  Dim intMemNo As Integer                          ' メモリ No.
  Dim blnMemNoDec As Boolean                       ' 設定値の折り返しフラグ
  Dim lngVIGDSR As Long                            ' VIG の DSR 値
  Dim blnRDYb As Boolean                           ' READY Out bit
  Dim blnSWEb As Boolean                           ' Sweep End Bit
  Dim lngDMMSTB As Long                            ' DMM の STB 値
  Dim lngRowNo As Long                             ' 測定データを格納する行位置
  Dim lngMeasCnt As Long                           ' 測定回数
  Dim intNowRepCnt As Integer                      ' 現繰り返し回数
  Dim intNowMemNo As Integer                       ' VIG の現メモリ No.


  lngRowNo = lngStaRow                             ' データ格納開始行

  ' デバイス (VIG, DMM) を開いて初期化 ( タイムアウト 10s)
  Call ibdev(intIFBoardAdr, intVIGAdr, 0, T10s, 1, 0, intVIGDes)
  Call ibdev(intIFBoardAdr, intDMMAdr, 0, T10s, 1, 0, intDMMDes)

  ' 送受信ごとにアドレス設定を行う
  Call ibconfig(intVIGDes, IbcUnAddr, 1)
  Call ibconfig(intDMMDes, IbcUnAddr, 1)

  ' デバイス・クリアおよびパラメータの初期化
  Call gpibwrt(intVIGDes, "C,*RST")
```

```
Call gpibwrt(intDMMDes, "C,*RST")


' VIG 設定
Call gpibwrt(intVIGDes, "MD4")                  ' メモリ・リコール・モード
Call gpibwrt(intVIGDes, "RCLR")                 ' メモリを初期化

' メモリ設定
dblGParam = dblVMin                             ' 開始（停止）電圧を設定
Do
  ' メモリ設定パラメータを作成して送信
  strSendCmd = "N" & CStr(intMemNo) & ",SOV" & CStr(dblGParam) & "V,P"
  Call gpibwrt(intVIGDes, strSendCmd)
  If blnMemNoDec = False Then                   ' 減算か否か？
      dblGParam = dblGParam + dblVStep          ' 現在値にステップ値を加算
    Else
      dblGParam = dblGParam - dblVStep          ' 現在値からステップ値を減算
  End If
  If dblGParam >= dblVMax Then                  ' 折返し ( 最大 ) 電圧を超えたか？
    blnMemNoDec = True                          ' 減算に切り替え
    dblGParam = dblVMax - dblVStep              ' 折返しから次の値をセット
  End If
  intMemNo = intMemNo + 1                       ' メモリ番号をインクリメント
Loop Until dblGParam < dblVMin And blnMemNoDec = True
                                                ' 減算で最小値を下回るまで

Call gpibwrt(intVIGDes, "SCM0")                 ' スキャン・モードをホールド
Call gpibwrt(intVIGDes, "SC0," & CStr(intMemNo - 1))
                                                ' スキャン範囲（0 ～終了 No.）
Call gpibwrt(intVIGDes, "*CLS")                 ' ステータスをクリア

' DMM 設定
Call gpibwrt(intDMMDes, "F1")                   ' 直流電圧測定ファンクション
Call gpibwrt(intDMMDes, "PR2")                  ' サンプリング・レートを Middle
Call gpibwrt(intDMMDes, "TRS3")                 ' トリガを BUS
Call gpibwrt(intDMMDes, "H0")                   ' ヘッダ OFF
Call gpibwrt(intDMMDes, "*CLS")                 ' ステータスをクリア

Call gpibwrt(intVIGDes, "OPR")                  ' VIG の出力を ON にする
For intNowRepCnt = 1 To intRepSet Step 1        ' 設定した繰り返し回数まで繰り返す
  Do
    ' VIG 発生・確認
    Call gpibwrt(intVIGDes, "DSR?")             ' DSR 問い合わせ
    Call gpibrd(intVIGDes, strRecvStr)          ' データ受信
    lngVIGDSR = CLng(strRecvStr)                ' DSR 値格納
    'DSR の各機能 bit と論理積
    blnRDYb = lngVIGDSR And 1024                ' READY Out bit
    blnSWEb = lngVIGDSR And 8192                ' Sweep End bit

    If blnRDYb = True Then                      ' VIG Ready
      Call gpibwrt(intVIGDes, "RCLM?")          ' 現メモリ No. 問い合わせ
      Call gpibrd(intVIGDes, strRecvStr)        ' データ受信
      ' メモリ No. 文字列の取出し
      intNowMemNo = CInt(Replace(strRecvStr, "RCLM", ""))
      Call gpibwrt(intVIGDes, "SOV?")           ' 電圧発生値問い合わせ
      Call gpibrd(intVIGDes, strRecvStr)        ' データ受信
```

```
      strRecvStr = Replace(strRecvStr, "SOV", "")
      ' シートのセルにメモリ No. と発生パラメータを格納
      ActiveSheet.Cells(lngRowNo, lngRepCntCol).value = intNowRepCnt
      ActiveSheet.Cells(lngRowNo, lngMemNoCol).value = intNowMemNo
      ActiveSheet.Cells(lngRowNo, lngGenerateVCol).value = strRecvStr

      ' DMM 測定動作
      lngMeasCnt = lngMeasCnt + 1                  ' 測定回数をインクリメント
      Call gpibwrt(intDMMDes, "*TRG")              ' トリガを送信、測定開始
      Do
        Call gpibwrt(intDMMDes, "*STB?")           ' STB 問い合わせ
        Call gpibrd(intDMMDes, strRecvStr)         ' データ受信
        lngDMMSTB = CLng(strRecvStr)               ' STB 値格納
        lngDMMSTB = lngDMMSTB And 16               ' Message Available bit との論理積
      Loop Until lngDMMSTB = 16                    ' 測定終了まで繰り返し
      Call gpibrd(intDMMDes, strRecvStr)           ' 測定値を受信

      ' シートのセルに電圧測定データを格納
      ActiveSheet.Cells(lngRowNo, lngVDataCol).value = strRecvStr

      Call gpibwrt(intDMMDes, "F5")                ' 直流測定ファンクション
      Call gpibwrt(intDMMDes, "*TRG")              ' トリガを送信、測定開始
      Do
        Call gpibwrt(intDMMDes, "*STB?")           ' STB 問い合わせ
        Call gpibrd(intDMMDes, strRecvStr)         ' データ受信
        lngDMMSTB = CLng(strRecvStr)               ' STB 値格納
        lngDMMSTB = lngDMMSTB And 16               ' Message Available bit との論理積
      Loop Until lngDMMSTB = 16                    ' 測定終了まで繰り返し
      Call gpibrd(intDMMDes, strRecvStr)           ' 測定値を受信

      ' シートのセルに測定回数と電流測定データを格納
      ActiveSheet.Cells(lngRowNo, lngMeasNoCol).value = lngMeasCnt
      ActiveSheet.Cells(lngRowNo, lngIDataCol).value = strRecvStr


      Call gpibwrt(intDMMDes, "F1")                ' 直流電圧測定ファンクション
      Call gpibwrt(intVIGDes, "*TRG")              ' VIG を次のメモリ No. へ
      lngRowNo = lngRowNo + 1                      ' 行番号をインクリメント

    End If
    DoEvents                                       ' 一時的に OS に制御を移す

    Loop Until blnSWEb = True                      ' Sweep End bit が立つまで繰り返し
  Next


  Call gpibwrt(intVIGDes, "SBY")                   ' VIG の出力を OFF にする

  Call ibonl(intVIGDes, 0)                         ' VIG をオフラインにする
  Call ibonl(intDMMDes, 0)                         ' DMM をオフラインにする

End Sub
```

5.9.3.2 USB でのプログラム例

```
Option Explicit                                    ' すべての変数を明示的に宣言

' 機器・インタフェース設定
Private Const lngVIGID As Long = 1                 ' VIG の USB_ID...1
Private Const lngDMMID As Long = 2                 ' DMM の USB_ID...2
Private Const lngTimeOut As Long = 10              ' USB タイムアウト時間 [s]

' 動作設定パラメータ
Private Const dblVMin As Double = 0                ' 開始（最小）電圧 [V]
Private Const dblVMax As Double = 5                ' 折返し（最大）電圧 [V]
Private Const dblVStep As Double = 0.1             ' 電圧ステップ値 [V]
Private Const intRepSet As Integer = 2             ' 繰り返し回数

' データ格納するシートの設定
Private Const lngStaRow As Long = 1                ' データを入れる開始行
Private Const lngRepCntCol As Long = 1             ' 繰り返し回数の桁
Private Const lngMeasNoCol As Long = 2             ' 測定回数の桁
Private Const lngMemNoCol As Long = 3              ' メモリ No. の桁
Private Const lngGenerateVCol As Long = 4          ' 発生電圧値の桁
Private Const lngVDataCol As Long = 5              ' 測定電圧の桁
Private Const lngIDataCol As Long = 6              ' 測定電流の桁

Public blnUSBComErr As Boolean                     ' USB送受信のエラー発生状態
Private Const OK As Integer = 0                    ' 定数「OK」を宣言


Public Sub SetMemAndMeasure_USB()

  Dim lngVIGHdl As Long                            ' VIG の USB ハンドル
  Dim lngDMMHdl As Long                            ' DMM の USB ハンドル
  Dim strRecvStr As String                         ' データ受信用バッファ
  Dim strSendCmd As String                         ' メモリ設定用コマンド文字列
  Dim dblGParam As Double                          ' 発生値パラメータ
  Dim intMemNo As Integer                          ' メモリ No.
  Dim blnMemNoDec As Boolean                       ' 設定値の折り返しフラグ
  Dim lngVIGDSR As Long                            ' VIG の DSR 値
  Dim blnRDYb As Boolean                           ' READY Out Bit
  Dim blnSWEb As Boolean                           ' Sweep End Bit
  Dim lngDMMSTB As Long                            ' DMM の STB 値
  Dim lngRowNo As Long                             ' 測定データを格納する行位置
  Dim lngMeasCnt As Long                           ' 測定回数
  Dim intNowRepCnt As Integer                      ' 現繰り返し回数
  Dim intNowMemNo As Integer                       ' VIG の現メモリ No.


  lngRowNo = lngStaRow                             ' データ格納開始行

  Call aUSBSta(lngTimeOut)                         ' USB ドライバ初期化

  ' デバイス（機器）のオープンとハンドルの取得
  Call aUSBOpn(lngVIGHdl, lngVIGID)                ' VIG をオープン
  Call aUSBOpn(lngDMMHdl, lngDMMID)                ' DMM をオープン
  If blnUSBComErr = True Then                      ' USB エラー確認
    GoTo ErrExit
```

```
  End If

' デバイス・クリアおよびパラメータの初期化
Call aUSBWrt(lngVIGHdl, "C,*RST")
Call aUSBWrt(lngDMMHdl, "C,*RST")

' VIG 設定
Call aUSBWrt(lngVIGHdl, "MD4")                      ' メモリ・リコール・モード
Call aUSBWrt(lngVIGHdl, "RCLR")                     ' メモリを初期化

' メモリ設定
dblGParam = dblVMin                                 ' 開始 ( 停止 ) 電圧を設定
Do
  ' メモリ設定パラメータを作成して送信
  strSendCmd = "N" & CStr(intMemNo) & ",SOV" & CStr(dblGParam) & "V,P"
  Call aUSBWrt(lngVIGHdl, strSendCmd)
  If blnMemNoDec = False Then                       ' 減算か否か？
      dblGParam = dblGParam + dblVStep              ' 現在値にステップ値を加算
    Else
      dblGParam = dblGParam - dblVStep              ' 現在値からステップ値を減算
  End If
  If dblGParam >= dblVMax Then                      ' 折返し ( 最大 ) 電圧を超えたか？
    blnMemNoDec = True                              ' 減算に切り替え
    dblGParam = dblVMax - dblVStep                  ' 折返しから次の値をセット
  End If
  intMemNo = intMemNo + 1                           ' メモリ No. をインクリメント
  If blnUSBComErr = True Then                       ' USB エラー確認
    GoTo ErrExit
  End If
Loop Until dblGParam < dblVMin And blnMemNoDec = True
                                                    ' 減算で最小値を下回るまで
Call aUSBWrt(lngVIGHdl, "SCM0")                     ' スキャン・モードをホールド
Call aUSBWrt(lngVIGHdl, "SC0," & CStr(intMemNo - 1))
                                                    ' スキャン範囲（0 ～終了 No.）
Call aUSBWrt(lngVIGHdl, "*CLS")                     ' ステータスをクリア

' DMM 設定
Call aUSBWrt(lngDMMHdl, "F1")                       ' 直流電圧測定ファンクション
Call aUSBWrt(lngDMMHdl, "PR2")                      ' サンプリング・レートを Middle
Call aUSBWrt(lngDMMHdl, "TRS3")                     ' トリガを BUS
Call aUSBWrt(lngDMMHdl, "H0")                       ' ヘッダ OFF
Call aUSBWrt(lngDMMHdl, "*CLS")                     ' ステータスをクリア

' VIG の設定動作終了を待つ
Do
  Call aUSBWrt(lngVIGHdl, "*OPC?")                  ' 全動作終了？
  Call aUSBRd(lngVIGHdl, strRecvStr)                ' データ受信

  ' USB 送受信にエラーないか確認
  If blnUSBComErr = True Then
    GoTo ErrExit
  End If
Loop Until strRecvStr = "1"

' DMM の設定動作終了を待つ
Do
  Call aUSBWrt(lngDMMHdl, "*OPC?")                  ' 全動作終了？
  Call aUSBRd(lngDMMHdl, strRecvStr)                ' データ受信
```

```
        ' USB 送受信にエラーないか確認
        If blnUSBComErr = True Then
          GoTo ErrExit
        End If
      Loop Until strRecvStr = "1"

      Call aUSBWrt(lngVIGHdl, "OPR")                    ' VIG の出力を ON にする

      For intNowRepCnt = 1 To intRepSet Step 1
        Do
          ' VIG 発生・確認
          Call aUSBWrt(lngVIGHdl, "DSR?")               ' DSR 問い合わせ
          Call aUSBRd(lngVIGHdl, strRecvStr)            ' データ受信
          lngVIGDSR = CLng(strRecvStr)                  ' DSR 値格納

          'DSR の各機能 bit と論理積
          blnRDYb = lngVIGDSR And 1024                  ' READY Out bit
          blnSWEb = lngVIGDSR And 8192                  ' Sweep End bit

          If blnUSBComErr = True Then                   ' USB エラー確認
            GoTo ErrExit
          End If

          If blnRDYb = True Then                        ' VIG Ready
            Call aUSBWrt(lngVIGHdl, "RCLM?")            ' 現メモリ No. 問い合わせ
            Call aUSBRd(lngVIGHdl, strRecvStr)          ' データ受信
            ' メモリ No. 文字列の取出し
            intNowMemNo = CInt(Replace(strRecvStr, "RCLM", ""))

            Call aUSBWrt(lngVIGHdl, "SOV?")             ' 電圧発生値問い合わせ
            Call aUSBRd(lngVIGHdl, strRecvStr)          ' データ受信
            strRecvStr = Replace(strRecvStr, "SOV", "")
            ' シートのセルにメモリ No. と発生パラメータを格納
            ActiveSheet.Cells(lngRowNo, lngRepCntCol).Value = intNowRepCnt
            ActiveSheet.Cells(lngRowNo, lngMemNoCol).Value = intNowMemNo
            ActiveSheet.Cells(lngRowNo, lngGenerateVCol).Value = strRecvStr

            ' DMM 測定動作
            lngMeasCnt = lngMeasCnt + 1                 ' 測定回数をインクリメント
            Call aUSBWrt(lngDMMHdl, "*TRG")             ' トリガを送信、測定開始
            Do
              Call aUSBWrt(lngDMMHdl, "*STB?")          ' STB 問い合わせ
              Call aUSBRd(lngDMMHdl, strRecvStr)        ' データ受信
              lngDMMSTB = CLng(strRecvStr)              ' STB 値格納
              lngDMMSTB = lngDMMSTB And 16              ' Message Available bit との論理積
              If blnUSBComErr = True Then               ' USB エラー確認
                GoTo ErrExit
              End If
            Loop Until lngDMMSTB = 16                   ' 測定終了まで繰り返し
            Call aUSBRd(lngDMMHdl, strRecvStr)          ' 測定値を受信

            ' シートのセルに電圧測定データを格納
            ActiveSheet.Cells(lngRowNo, lngVDataCol).Value = strRecvStr

            Call aUSBWrt(lngDMMHdl, "F5")               ' 直流測定ファンクション
            Call aUSBWrt(lngDMMHdl, "*TRG")             ' トリガを送信、測定開始
```

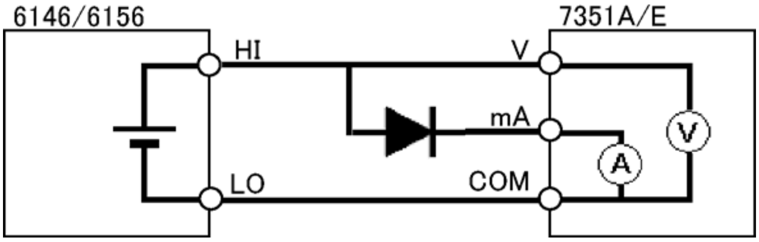
```
        Do
          Call aUSBWrt(lngDMMHdl, "*STB?")          ' STB 問い合わせ
          Call aUSBRd(lngDMMHdl, strRecvStr)        ' データ受信
          lngDMMSTB = CLng(strRecvStr)              ' STB 値格納
          lngDMMSTB = lngDMMSTB And 16              ' Message Available bit との論理積
          If blnUSBComErr = True Then               ' USB エラー確認
            GoTo ErrExit
          End If
        Loop Until lngDMMSTB = 16                   ' 測定終了まで繰り返し
        Call aUSBRd(lngDMMHdl, strRecvStr)          ' 測定値を受信

        ' シートのセルに測定回数と電流測定データを格納
        ActiveSheet.Cells(lngRowNo, lngMeasNoCol).Value = lngMeasCnt
        ActiveSheet.Cells(lngRowNo, lngIDataCol).Value = strRecvStr

        Call aUSBWrt(lngDMMHdl, "F1")               ' 直流電圧測定ファンクション
        Call aUSBWrt(lngVIGHdl, "*TRG")             ' VIG を次のメモリ No. へ

        If blnUSBComErr = True Then                 ' USB エラー確認
          GoTo ErrExit
        End If
        lngRowNo = lngRowNo + 1                     ' 行番号をインクリメント
      End If
      DoEvents                                      ' 一時的に OS に制御を移す

      Loop Until blnSWEb = True                     ' Sweep End bit が立つまで繰り返し
    Next


ErrExit:

    Call aUSBWrt(lngVIGHdl, "SBY")                  ' VIG の出力を OFF にする

    ' デバイス（機器）のクローズ
    Call aUSBCls(lngVIGHdl)                         ' VIG
    Call aUSBCls(lngDMMHdl)                         ' DMM

    ' USB ドライバの終了
    Call aUSBEnd

End Sub
```

5.9.4 プログラム例 3- スイープ動作、DMM での電圧・電流の測定

本器をスイープ発生モード、シングル・スイープにて、0V ~ 3V まで 10mV ステップで直流電圧を発生させ、電圧・電流を DMM で計測します。（ダイオードの電圧 - 電流特性の測定）

ダイオードの破壊防止のため、電流リミットを 10mA に設定します。
測定回数、リミッタ作動検知、発生値、計測値を Excel シートのセルに格納します。

接続図



ファンクション :	電圧発生
開始値 :	0V
終了値 :	3V
ステップ値 :	10mV
ステップ時間 :	500ms
スイープ・モード :	シングル
電流リミット :	10mA
センス状態 :	内部（2W）

使用 DMM: 7351A/E

測定ファンクション :	直流電圧、直流電流
測定レンジ :	オート
サンプリング・レート :	MED（積分時間 : 20ms）
ヘッダ :	OFF

注意 DMM **の測定トリガはプログラムで** VIG **のステータスを確認して行っていますが、シングル・スイープでは** DMM **の測定完了を待たないので、コンピュータの負荷状態によってはスイープ発生と測定の同期が外れることがあります。**

（実行結果例）

（測定回数）	（リミッタ作動）	（発生電圧値）	（測定電圧値）	（測定電流値）
1		+0.00000E+0	-000.330E-03	-000.004E-03
2		+0.01000E+0	+009.730E-03	-000.001E-03
3		+0.02000E+0	+019.789E-03	+000.000E-03
.		.	.	.
省略		.	.	.
.		.	.	.
211	LMH	+2.10000E+0	+02.1110E+00	+010.132E-03
.		.	.	.
省略		.	.	.
.		.	.	.
300		+2.99000E+0	+02.1110E+00	+010.132E-03
301		+3.00000E+0	+02.1110E+00	+010.132E-03

5.9.4.1 GPIB でのプログラム例

```
Option Explicit                                  ' すべての変数を明示的に宣言

Private Const intIFBoardAdr As Integer = 0       ' GPIB ボードのアドレスを 0
Private Const intVIGAdr As Integer = 1           ' VIG のアドレスを 1
Private Const intDMMAdr As Integer = 2           ' DMM のアドレスを 2
Private Const lngStaRow As Long = 1              ' データを入れる開始行
Private Const lngMeaNoCol As Long = 1            ' 測定 No. を入れる桁
Private Const lngSrcLimitCol As Long = 2         ' VIG リミッタ作動通知の桁
Private Const lngSrcDataCol As Long = 3          ' VIG 発生値を入れる桁
Private Const lngVDataCol As Long = 4            ' 測定電圧データを入れる桁
Private Const lngIDataCol As Long = 5            ' 測定電流データを入れる桁

Public Sub SweepAndMeasure_GPIB()

  Dim intVIGDes As Integer                       ' VIG のデバイス・ディスクリプタ
  Dim intDMMDes As Integer                       ' DMM のデバイス・ディスクリプタ
  Dim strRecvStr As String                       ' データ受信用バッファ
  Dim strSrcLimit As String                      ' VIG リミッタ作動通知
  Dim strSrcData As String                       ' VIG 現在発生値
  Dim strMeasVData As String                     ' 測定電圧値
  Dim strMeasIData As String                     ' 測定電流値
  Dim lngMeasCnt As Long                         ' 測定回数
  Dim lngRowNo As Long                           ' 測定データを格納する行位置

  Dim lngDMMSTB As Long                          ' DMM の STB 値
  Dim lngVIGDSR As Long                          ' VIG の DSR 値
  Dim blnRDYb As Boolean                         ' READY Out Bit
  Dim blnSWEb As Boolean                         ' Sweep End Bit
  Dim blnLMHb As Boolean                         ' Limiter High Bit
  Dim blnLMLb As Boolean                         ' Limiter Low Bit

  lngRowNo = lngStaRow                           ' データ格納開始行

  ' デバイス (VIG, DMM) を開いて初期化 ( タイムアウト 10s)
  Call ibdev(intIFBoardAdr, intVIGAdr, 0, T10s, 1, 0, intVIGDes)
```

5.9.4 プログラム例 3- スイープ動作、DMM での電圧・電流の測定

```
Call ibdev(intIFBoardAdr, intDMMAdr, 0, T10s, 1, 0, intDMMDes)

' 送受信ごとにアドレス設定を行う
Call ibconfig(intVIGDes, IbcUnAddr, 1)
Call ibconfig(intDMMDes, IbcUnAddr, 1)

' デバイス・クリアおよびパラメータの初期化
Call gpibwrt(intVIGDes, "C,*RST")
Call gpibwrt(intDMMDes, "C,*RST")

' VIG 設定
Call gpibwrt(intVIGDes, "VF")                ' 電圧発生ファンクション
Call gpibwrt(intVIGDes, "MD2")               ' スイープ発生モード
Call gpibwrt(intVIGDes, "SCM1")              ' スイープ・モード・シングル
Call gpibwrt(intVIGDes, "SP0.5")             ' スイープ・ステップ時間 0.5s
Call gpibwrt(intVIGDes, "SN0,3,0.01")        ' 開始 0V, 終了 3V, ステップ 10mV
Call gpibwrt(intVIGDes, "SB0")               ' スイープ時バイアス 0V
Call gpibwrt(intVIGDes, "LMI 0.01")          ' 電流リミッタを 10mA
Call gpibwrt(intVIGDes, "ROD0")              ' READY 信号遅延を 10ms
Call gpibwrt(intVIGDes, "*CLS")              ' ステータスをクリア

' DMM 設定
Call gpibwrt(intDMMDes, "F1")                ' 直流電圧測定ファンクション
Call gpibwrt(intDMMDes, "PR2")               ' サンプリング・レートを Middle
Call gpibwrt(intDMMDes, "TRS3")              ' トリガを BUS
Call gpibwrt(intDMMDes, "H0")                ' ヘッダ OFF
Call gpibwrt(intDMMDes, "*CLS")              ' ステータスをクリア


Call gpibwrt(intVIGDes, "OPR")               ' VIG の出力を ON にする
' VIG のオペレート動作終了を待つ
Do
  Call gpibwrt(intVIGDes, "DSR?")            ' DSR 問い合わせ
  Call gpibrd(intVIGDes, strRecvStr)         ' データ受信
  lngVIGDSR = CLng(strRecvStr)               ' DSR 値格納
  lngVIGDSR = lngVIGDSR And 1024             ' READY Out bit との論理積
Loop Until lngVIGDSR = 1024                  ' READY Out bit が立つまで繰り返し

Call gpibwrt(intVIGDes, "*TRG")              ' スイープ開始

Do
  Call gpibwrt(intVIGDes, "DSR?")            ' DSR 問い合わせ
  Call gpibrd(intVIGDes, strRecvStr)         ' データ受信
  lngVIGDSR = CLng(strRecvStr)               ' DSR 値格納
  'DSR の各機能 bit と論理積
  blnRDYb = lngVIGDSR And 1024               ' READY Out bit
  blnSWEb = lngVIGDSR And 8192               ' Sweep End bit
  blnLMHb = lngVIGDSR And 128                ' Limiter High bit
  blnLMLb = lngVIGDSR And 64                 ' Limiter Low bit

  If blnLMHb = True Then                     ' Limiter High のチェック
    strSrcLimit = "LMH"
  End If

  If blnLMLb = True Then                     ' Limiter Low のチェック
    strSrcLimit = strSrcLimit & "LML"
  End If
```

```
    If blnRDYb = True Then
      lngMeasCnt = lngMeasCnt + 1                ' 測定回数をインクリメント
      Call gpibwrt(intDMMDes, "*TRG")            ' トリガを送信、測定開始
      Do
        Call gpibwrt(intDMMDes, "*STB?")         ' STB 問い合わせ
        Call gpibrd(intDMMDes, strRecvStr)       ' データ受信
        lngDMMSTB = CLng(strRecvStr)             ' STB 値格納
        lngDMMSTB = lngDMMSTB And 16             ' Message Available bit との論理積
      Loop Until lngDMMSTB = 16                  ' 測定終了まで繰り返し
      Call gpibrd(intDMMDes, strMeasVData)       ' DMM の測定値を受信

      Call gpibwrt(intDMMDes, "F5")              ' 直流電流測定ファンクション
      Call gpibwrt(intDMMDes, "*TRG")            ' トリガを送信、測定開始
      Do
        Call gpibwrt(intDMMDes, "*STB?")         ' STB 問い合わせ
        Call gpibrd(intDMMDes, strRecvStr)       ' データ受信
        lngDMMSTB = CLng(strRecvStr)             ' STB 値格納
        lngDMMSTB = lngDMMSTB And 16             ' Message Available bit との論理積
      Loop Until lngDMMSTB = 16                  ' 測定終了まで繰り返し
      Call gpibrd(intDMMDes, strMeasIData)       ' DMM の測定値を受信

      Call gpibrd(intVIGDes, strSrcData)         ' VIG の現在発生値を受信
      strSrcData = Replace(strSrcData, "SOV", "") ' ヘッダ文字列を削除

      ' シートのセルに各データを格納
      With ActiveSheet
        .Cells(lngRowNo, lngMeaNoCol).value = lngMeasCnt
        .Cells(lngRowNo, lngSrcLimitCol).value = strSrcLimit
        .Cells(lngRowNo, lngSrcDataCol).value = strSrcData
        .Cells(lngRowNo, lngVDataCol).value = strMeasVData
        .Cells(lngRowNo, lngIDataCol).value = strMeasIData
      End With
      strSrcLimit = ""

      Call gpibwrt(intDMMDes, "F1")              ' 直流電圧測定ファンクション
      lngRowNo = lngRowNo + 1                    ' 行番号をインクリメント
    End If
    DoEvents                                     ' 一時的に OS  に制御を移す

  Loop Until blnSWEb = True                      ' Sweep End bit が立つまで繰り返し

  Call gpibwrt(intVIGDes, "SBY")                 ' VIG の出力を OFF にする
  Call ibonl(intVIGDes, 0)                       ' VIG をオフラインにする
  Call ibonl(intDMMDes, 0)                       ' DMM をオフラインにする

End Sub
```

5.9.4.2 USB でのプログラム例

```
Option Explicit                                     ' すべての変数を明示的に宣言

Public blnUSBComErr As Boolean                      ' USB 送受信のエラー発生状態
Private Const OK As Integer = 0                     ' 定数「OK」を宣言
Private Const lngVIGID As Long = 1                  ' VIG の USB_ID...1
Private Const lngDMMID As Long = 2                  ' DMM の USB_ID...2
Private Const lngTimeOut As Long = 10               ' USB タイムアウト時間 [s]
Private Const lngStaRow As Long = 1                 ' データを入れる開始行
Private Const lngMeaNoCol As Long = 1               ' 測定 No. を入れる桁
Private Const lngSrcLimitCol As Long = 2            ' VIG リミッタ作動通知の桁
Private Const lngSrcDataCol As Long = 3             ' VIG 発生値を入れる桁
Private Const lngVDataCol As Long = 4               ' 測定電圧データを入れる桁
Private Const lngIDataCol As Long = 5               ' 測定電流データを入れる桁


Public Sub SweepAndMeasure_USB()

  Dim lngVIGHdl As Long                             ' VIG の USB ハンドル
  Dim lngDMMHdl As Long                             ' DMM の USB ハンドル
  Dim strRecvStr  As String                         ' データ受信用バッファ
  Dim strSrcLimit As String                         ' VIG リミッタ作動通知
  Dim strSrcData As String                          ' VIG 現在発生値
  Dim strMeasVData As String                        ' 測定電圧値
  Dim strMeasIData As String                        ' 測定電流値
  Dim lngMeasCnt As Long                            ' 測定回数
  Dim lngRowNo As Long                              ' 測定データを格納する行位置

  Dim lngDMMSTB As Long                             ' DMM の STB 値
  Dim lngVIGDSR As Long                             ' VIG の DSR 値
  Dim blnRDYb As Boolean                            ' READY Out Bit
  Dim blnSWEb As Boolean                            ' Sweep End Bit
  Dim blnLMHb As Boolean                            ' Limiter High Bit
  Dim blnLMLb As Boolean                            ' Limiter Low Bit

  lngRowNo = lngStaRow                              ' データ格納開始行

  Call aUSBSta(lngTimeOut)                          ' USB ドライバ初期化

  ' デバイス（機器）のオープンとハンドルの取得
  Call aUSBOpn(lngVIGHdl, lngVIGID)                 ' VIG をオープン
  Call aUSBOpn(lngDMMHdl, lngDMMID)                 ' DMM をオープン
  If blnUSBComErr = True Then                       ' USB エラー確認
    GoTo ErrExit
  End If

  ' デバイス・クリアおよびパラメータの初期化
  Call aUSBWrt(lngVIGHdl, "C,*RST")
  Call aUSBWrt(lngDMMHdl, "C,*RST")

  ' VIG 設定
  Call aUSBWrt(lngVIGHdl, "VF")                     ' 電圧発生ファンクション
  Call aUSBWrt(lngVIGHdl, "MD2")                    ' スイープ発生モード
  Call aUSBWrt(lngVIGHdl, "SCM1")                   ' スイープ・モード・シングル
  Call aUSBWrt(lngVIGHdl, "SP0.5")                  ' スイープ・ステップ時間 0.5s
```

```
Call aUSBWrt(lngVIGHdl, "SN0,3,0.01")          ' 開始 0V, 終了 3V, ステップ 10mV
Call aUSBWrt(lngVIGHdl, "SB0")                 ' スイープ時バイアス 0V
Call aUSBWrt(lngVIGHdl, "LMI 0.01")            ' 電流リミッタを 10mA
Call aUSBWrt(lngVIGHdl, "ROD0")                ' READY 信号遅延を 10ms
Call aUSBWrt(lngVIGHdl, "*CLS")                ' ステータスをクリア

' DMM 設定
Call aUSBWrt(lngDMMHdl, "F1")                  ' 直流電圧測定ファンクション
Call aUSBWrt(lngDMMHdl, "PR2")                 ' サンプリング・レートを Middle
Call aUSBWrt(lngDMMHdl, "TRS3")                ' トリガを BUS
Call aUSBWrt(lngDMMHdl, "H0")                  ' ヘッダ OFF
Call aUSBWrt(lngDMMHdl, "*CLS")                ' ステータスをクリア

' VIG の設定動作終了を待つ
Do
  Call aUSBWrt(lngVIGHdl, "*OPC?")             ' 全動作終了？
  Call aUSBRd(lngVIGHdl, strRecvStr)           ' データ受信

  ' USB 送受信にエラーないか確認
  If blnUSBComErr = True Then
    GoTo ErrExit
  End If
Loop Until strRecvStr = "1"

' DMM の設定動作終了を待つ
Do
  Call aUSBWrt(lngDMMHdl, "*OPC?")             ' 全動作終了？
  Call aUSBRd(lngDMMHdl, strRecvStr)           ' データ受信

  ' USB 送受信にエラーないか確認
  If blnUSBComErr = True Then
    GoTo ErrExit
  End If
Loop Until strRecvStr = "1"

Call aUSBWrt(lngVIGHdl, "OPR")                 ' VIG の出力を ON にする
' VIG のオペレート動作終了を待つ
Do
  Call aUSBWrt(lngVIGHdl, "DSR?")              ' DSR 問い合わせ
  Call aUSBRd(lngVIGHdl, strRecvStr)           ' データ受信
  lngVIGDSR = CLng(strRecvStr)                 ' DSR 値格納
  blnRDYb = lngVIGDSR And 1024                 ' READY Out bit との論理積

  ' USB 送受信にエラーないか確認
  If blnUSBComErr = True Then
    GoTo ErrExit
  End If
Loop Until blnRDYb = True                      ' READY Out bit が立つまで繰り返し

Call aUSBWrt(lngVIGHdl, "*TRG")                ' スイープ開始

Do
  Call aUSBWrt(lngVIGHdl, "DSR?")              ' DSR 問い合わせ
  Call aUSBRd(lngVIGHdl, strRecvStr)           ' データ受信

  If blnUSBComErr = True Then                  ' USB 送受信エラー確認
```

```
  GoTo ErrExit
End If
lngVIGDSR = CLng(strRecvStr)                  ' DSR 値格納
'DSR の各機能 bit と論理積
blnRDYb = lngVIGDSR And 1024                  ' READY Out bit
blnSWEb = lngVIGDSR And 8192                  ' Sweep End bit
blnLMHb = lngVIGDSR And 128                   ' Limiter High bit
blnLMLb = lngVIGDSR And 64                    ' Limiter Low bit

If blnLMHb = True Then                        ' Limiter High のチェック
  strSrcLimit = "LMH"
End If
If blnLMLb = True Then                        ' Limiter Low のチェック
  strSrcLimit = strSrcLimit & "LML"
End If

If blnRDYb = True Then
  lngMeasCnt = lngMeasCnt + 1                 ' 測定回数をインクリメント
  Call aUSBWrt(lngDMMHdl, "*TRG")             ' トリガを送信、測定開始
  Do
    Call aUSBWrt(lngDMMHdl, "*STB?")          ' STB 問い合わせ
    Call aUSBRd(lngDMMHdl, strRecvStr)        ' データ受信
    lngDMMSTB = CLng(strRecvStr)              ' STB 値格納
    lngDMMSTB = lngDMMSTB And 16              ' Message Available bit との論理積
    If blnUSBComErr = True Then               ' USB 送受信エラー確認
      GoTo ErrExit
    End If
  Loop Until lngDMMSTB = 16                   ' 測定終了まで繰り返し
Call aUSBRd(lngDMMHdl, strMeasVData)          ' DMM の測定値を受信

Call aUSBWrt(lngDMMHdl, "F5")                 ' 直流電流測定ファンクション
Call aUSBWrt(lngDMMHdl, "*TRG")               ' トリガを送信、測定開始
Do
  Call aUSBWrt(lngDMMHdl, "*STB?")            ' STB 問い合わせ
  Call aUSBRd(lngDMMHdl, strRecvStr)          ' データ受信
  lngDMMSTB = CLng(strRecvStr)                ' STB 値格納
  lngDMMSTB = lngDMMSTB And 16                ' Message Available bit との論理積
  If blnUSBComErr = True Then                 ' USB 送受信エラー確認
    GoTo ErrExit
  End If
  Loop Until lngDMMSTB = 16                   ' 測定終了まで繰り返し
  Call aUSBRd(lngDMMHdl, strMeasIData)        ' DMM の測定値を受信

  Call aUSBRd(lngVIGHdl, strSrcData)          ' VIG の現在発生値を受信
  strSrcData = Replace(strSrcData, "SOV", "") ' ヘッダ文字列を削除

  ' シートのセルに各データを格納
  With ActiveSheet
    .Cells(lngRowNo, lngMeaNoCol).Value = lngMeasCnt
    .Cells(lngRowNo, lngSrcLimitCol).Value = strSrcLimit
    .Cells(lngRowNo, lngSrcDataCol).Value = strSrcData
    .Cells(lngRowNo, lngVDataCol).Value = strMeasVData
    .Cells(lngRowNo, lngIDataCol).Value = strMeasIData
  End With
  strSrcLimit = ""

  Call aUSBWrt(lngDMMHdl, "F1")               ' 直流電圧測定ファンクション
```

```
      lngRowNo = lngRowNo + 1                         ' 行番号をインクリメント

    End If

    DoEvents                                          ' 一時的に OS に制御を移す

  Loop Until blnSWEb = True                           ' Sweep End bit が立つまで繰り返し

ErrExit:

  Call aUSBWrt(lngVIGHdl, "SBY")                      ' VIG の出力を OFF にする

  ' デバイス（機器）のクローズ
  Call aUSBCls(lngVIGHdl)                             ' VIG
  Call aUSBCls(lngDMMHdl)                             ' DMM

  ' USB ドライバの終了
  Call aUSBEnd

End Sub
```